# AMD SB600 Register Reference Manual (Public Version)

**Technical Reference Manual**
**Rev. 3.03**

**P/N: 46155_sb600_rrg_pub_3.03**

**Trademarks**

AMD, the AMD Arrow logo, Athlon, and combinations thereof, ATI, ATI logo, Radeon, and Crossfire are trademarks of Advanced Micro Devices, Inc.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Microsoft and Windows are registered trademarks and Windows Vista is trademark of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

**Disclaimer**

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel, or otherwise, to any intellectual property rights are granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 About this Manual

This manual is a register reference guide for the AMD SB600 Southbridge. It integrates the key I/O, communications, and audio features required in a state-of-the-art PC into a single device. It is specifically designed to operate with AMD's RADEON IGP Xpress family of integrated graphics processor products in both desktop and mobile PCs.

## 1.2 Nomenclature and Conventions

### *1.2.1 Recent Updates*

Updates recent to each revision are highlighted in red.

### *1.2.2 Numeric Representations*

- Hexadecimal numbers are prefixed with "0x" or suffixed with "h," whenever there is a possibility of confusion. Other numbers are decimal.
- Registers (or fields) of an identical function are sometimes indicated by a single expression in which the part of the signal name that changes is enclosed in square brackets. For example, registers HOST_DATA0 through to HOST_DATA7 is represented by the single expression HOST_DATA[7:0].

### *1.2.3 Register Description*

All registers in this document are described with the format of the sample table below. All offsets are in hexadecimal notation, while programmed bits are in either binomial or hexadecimal notation.

**Table 1-1: Register Description Table Notation—Example**

| Latency Timer – RW – 8 bits – [Offset: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer (R/W) | 7:0 | 00h | This bit field is used to specify the time in number of PCI clocks, the SATA controller as a master is still allowed to control the PCI bus after its GRANT_L is deasserted. The lower three bits [0A:08] are hardwired to 0 h , resulting in a time granularity of 8 clocks. |

Latency Timer. Reset Value: 00h

| Register Information | Value/Content in the Example |
|---|---|
| Register name | Latency Timer |
| Read / Write capability<br>R = Readable<br>W = Writable<br>RW = Readable and Writable | RW |
| Register size | 8 bits |
| Register address(es)* | Offset: 0Dh |
| Field name | Latency Timer (R/W) |
| Field position/size | 7:0 |
| Field default value | 00h |
| Field description | “This bit … 8 clocks.” |
| Field mirror information | |
| Brief register description | Latency Timer. Reset Value: 00h |

*** Note**: There maybe more than one address; the convention used is as follows:

**[aperName:offset]** - single mapping, to one aperture/decode and one offset
**[aperName1, aperName2, …, aperNameN:offset]** - multiple mappings to different apertures/decodes but same offset
**[aperName:startOffset-endOffset]** - mapped to an offset range in the same aperture/decode

**Warning:** Do not attempt to modify values of registers or bit fields marked "Reserved." Doing so may cause the system to behave in unexpected manners.

## 1.3 Features of the SB600

**CPU Interface**

- Supports both Single and Dual core AMD CPUs
  - Desktop: Athlon 64, Athlon 64 FX, Athlon 64 X2, Sempron, Opteron, dual-core Opteron
  - Mobile: Athlon XP-M, Mobile Athlon 64, Turion 64, Mobile Sempron

**PCI Host Bus Controller**

- Supports PCI Rev. 2.3 specification
- Supports PCI bus at 33MHz
- Supports up to 6 bus master devices
- Supports 40-bit addressing
- Supports interrupt steering for plug-n-play devices
- Supports concurrent PCI operations
- Supports hiding of PCI devices by BIOS/hardware
- Supports spread spectrum on PCI clocks

**USB controllers**

- 5 OHCI and 1 EHCI Host controllers to support 10 USB ports
- All 10 ports are USB 1.1 ("Low Speed", "Full Speed") and 2.0 ("High Speed") compatible
- Supports ACPI S1~S5
- Supports legacy keyboard/mouse
- Supports USB debug port
- Supports port disable with individual control

**SMBus Controller**

- SMBus Rev. 2.0 compliant
- Support SMBALERT # signal / GPIO

**Interrupt Controller**

- Supports IOAPIC/X-IO APIC mode for 24 channels of interrupts
- Supports 8259 legacy mode for 15 interrupts
- Supports programmable level/edge triggering on each channels
- Supports serial interrupt on quiet and continuous modes

**DMA Controller**

- Two cascaded 8237 DMA controllers
- Supports PC/PCI DMA
- Supports LPC DMA
- Supports type F DMA

**LPC host bus controller**

- Supports LPC based super I/O and flash devices
- Supports two master/DMA devices
- Supports TPM version 1.1/1.2 devices for enhanced security
- Supports SPI devices

**SATA II AHCI Controller**

- Supports four SATA ports, complying with the SATA 2.0 specification
- Supports SATA II 3.0GHz PHY, with backward compatibility with 1.5GHz
- Supports RAID striping (RAID 0) across all 4 ports
- Supports RAID mirroring (RAID 1) across all 4 ports
- Supports RAID 10 (4 ports needed)
- Supports both AHCI mode and IDE mode
- Supports advanced power management with ACHI mode

**IDE Controller**

- Single PATA channel support
- Supports PIO, Multi-word DMA, and Ultra DMA 33/66/100/133 modes
- 32x32byte buffers on each channel for buffering
- Swap bay support by tri-state IDE signals
- Supports Message Signaled Interrupt (MSI)
- Integrated IDE series resistors

**AC Link interface**

- Supports for both audio and modem codecs
- Compliant with AC-97 codec Rev. 2.3
- 6/8 channel support on audio codec
- Multiple functions for audio and modem Codec operations
- Bus master logic
- Supports up to 3 codecs simultaneously
- Supports SPDIF output
- Separate bus from the HD audio

**HD Audio**

- 4 Independent output streams (DMA)
- 4 Independent input streams (DMA)
- Up to 16 channels of audio output per stream
- Supports up to 4 codecs
- Up to 192kHz sample rate
- Up to 32-bit per sample
- Message Signaled Interrupt (MSI) capability
- 64-bit addressing capability for MSI
- 64-bit addressing capability for DMA bus master
- Unified Audio Architecture (UAA) compatible
- HD Audio registers can be located anywhere in the 64-bit address space

**Timers**

- 8254-compatible timer
- Microsoft High Precision Event Timer (HPET)
- ACPI power management timer

**RTC (Real Time Clock)**

- 256-byte battery-backed CMOS RAM
- Hardware supported century rollover
- RTC battery monitoring feature

**Power Management**

- ACPI specification 2.0 compliant power management schemes
  - Supports C2, C3, C4, ACPI states
  - Supports C1e and C3 pop-up
  - Supports S0, S1, S2, S3, S4, and S5
- Wakeup events for S1, S2, S3, S4/S5 generated by:
  - Any GEVENT pin
  - Any GPM pin
  - USB
  - Power button
  - Internal RTC wakeup
  - SMI# event
- Full support for On-Now™
- Supports CPU SMM, generating SMI# signal upon power management events
- GPIO supports on external wake up events
- Supports CLKRUN# on PCI power management
- Provides clock generator and CPU STPCLK# control
- Support for ASF

## 1.4 Block Diagrams

This section contains two block diagrams for the SB600. *Figure 1* shows the SB600 internal PCI devices with their assigned bus, device, and function numbers. *Figure 2* shows the SB600 internal PCI devices and the major function blocks.

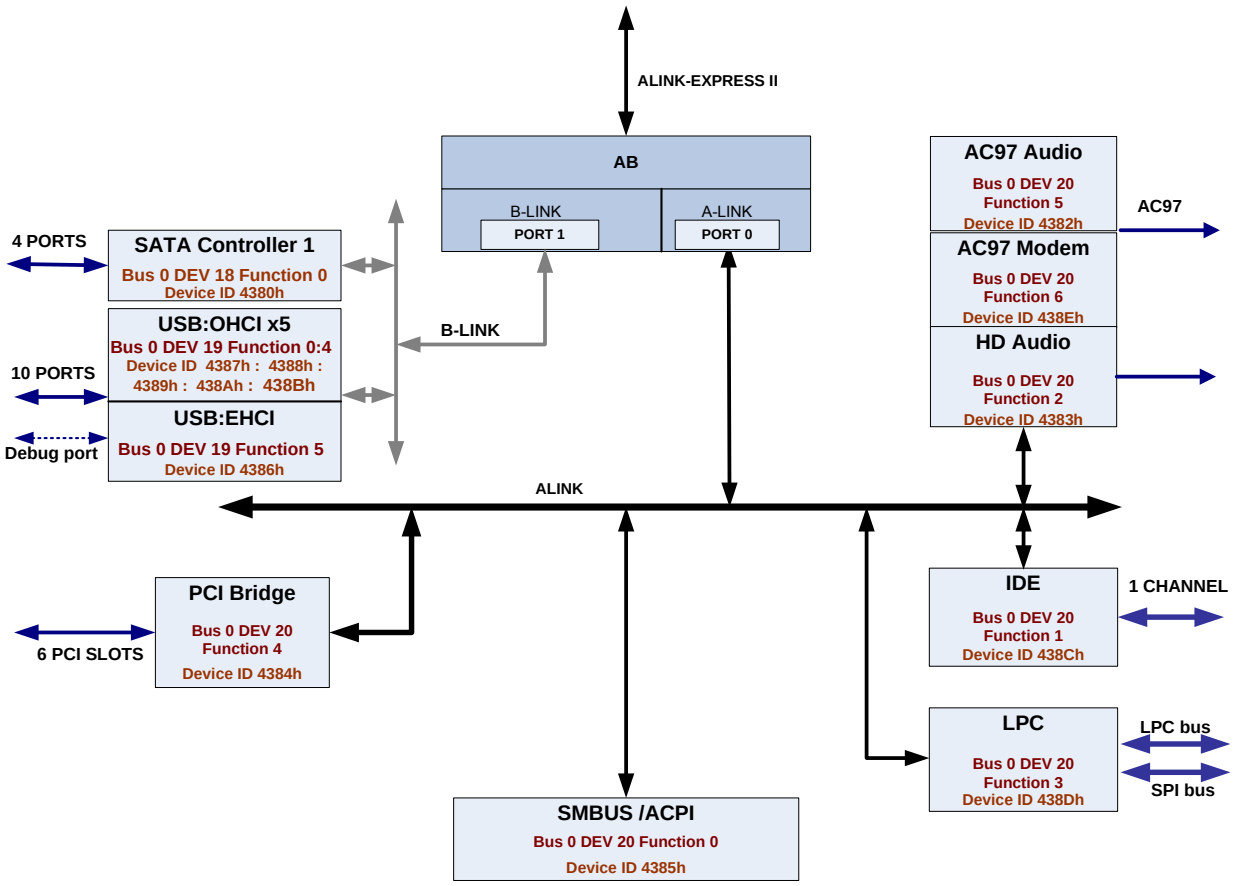



**Figure 1 SB600 PCI Internal Devices**

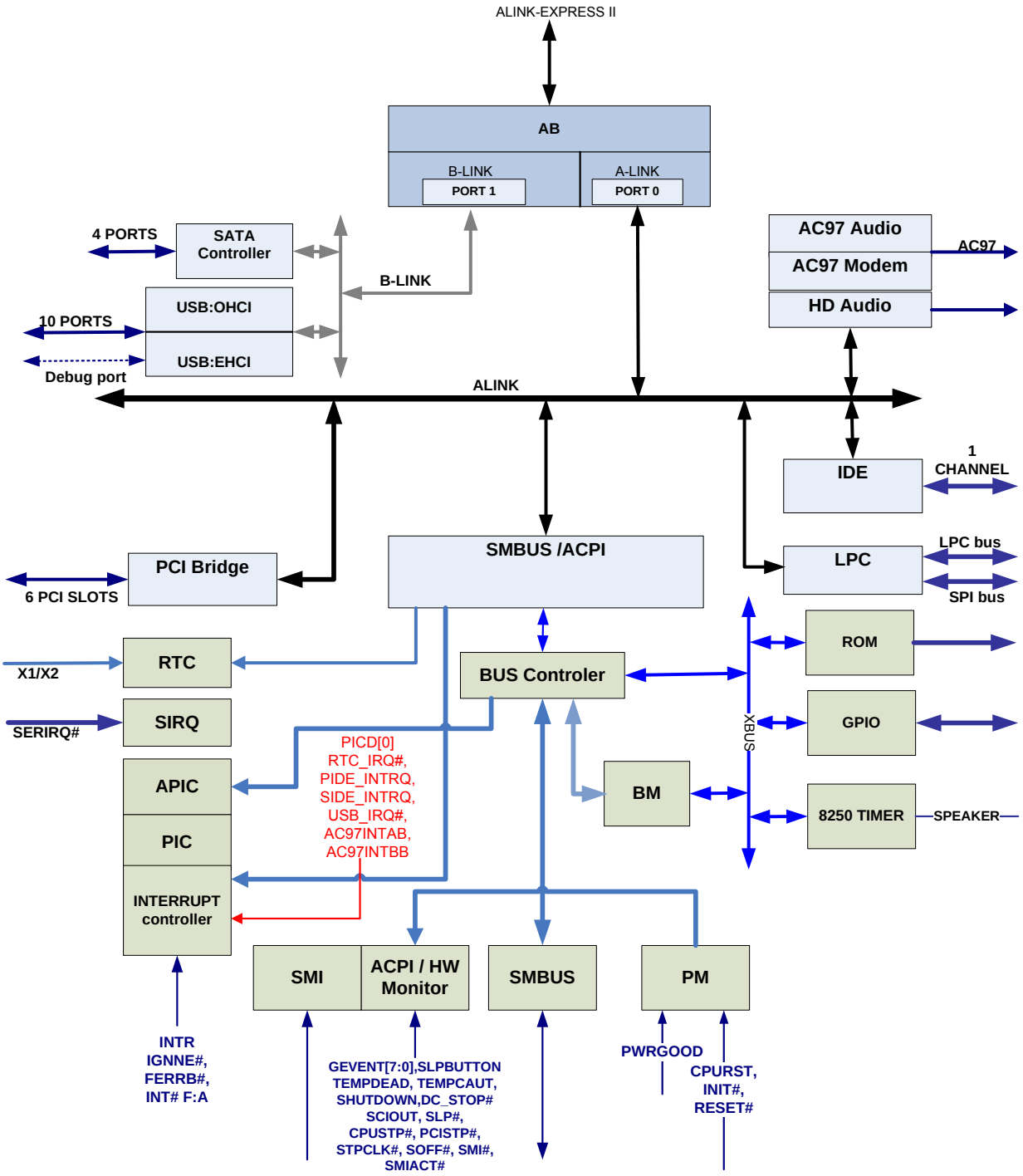
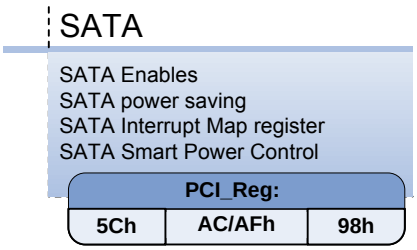



**Figure 2 SB600 PCI Internal Devices and Major Function Blocks**

# 2 Register Descriptions: PCI Devices

## 2.1 SATA Registers (Device 18, Function 0)

Note: Some SATA functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these SATA functions and the associated registers.




### *2.1.1 PCI Configuration Space*

The PCI Configuration Space registers define the operation of the SB600's SATA controller on the PCI bus. These registers are accessible only when the SATA controller detects a Configuration Read or Write operation, with its IDSEL asserted, on the 32-bit PCI bus.

| Register Name | Offset Address |
|---|---|
| Vendor ID | 00h |
| Device ID | 02h |
| Command | 04h |
| Status | 06h |
| Revision ID/Class Code | 08h |
| Cache Link Size | 0Ch |
| Master Latency Timer | 0Dh |
| Header Type | 0Eh |
| BIST Mode Type | 0Fh |
| Base Address 0 | 10h |
| Base Address 1 | 14h |
| Base Address 2 | 18h |
| Base Address 3 | 1Ch |
| Bus Master Interface Base Address | 20h |
| AHCI Base Address | 24h |
| Subsystem ID and Subsystem Vendor ID | 2Ch |
| Capabilities Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Min_gnt | 3Eh |
| Max_latency | 3Fh |
| Misc control | 40h |
| Watch Dog Control And Status | 44h |
| Watch Dog Counter | 46h |
| MSI Control | 50h |
| MSI Address | 54h |
| MSI Upper Address | 58h |
| MSI Data | 5Ch |
| Power Management Capability ID | 60h |
| Power Management Capability | 62h |
| Power Management Control And Status | 64h |

| Register Name | Offset Address |
|---|---|
| Serial ATA Capability Register 0 | 70h |
| Serial ATA Capability Register 1 | 74h |
| IDP Index | 78h |
| IDP Data | 7Ch |
| PHY Port0 Control | 88h |
| PHY Port1 Control | 8Ch |
| PHY Port2 Control | 90h |
| PHY Port3 Control | 94h |
| BIST pattern Count | C0h |
| PCI Target Control TimeOut Counter | C4h |

| Vendor ID - R - 16 bits - [PCI_Reg:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | This register holds a unique 16-bit value assigned to a vendor. Combined with the device ID, it identifies any PCI device. |

| Device ID - R - 16 bits - [PCI_Reg:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID | 15:0 | 4380h | This register holds a unique 16-bit value assigned to a device. Combined with the vendor ID, it identifies any PCI device.<br>4380h for the non-Raid5 controller<br>4381h for the Raid5 controller.<br>Bonding option default to the non-Raid 5 controller. |

| Command - RW - 16 bits - [PCI_Reg:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I/O Access Enable | 0 | 0b | This bit controls access to the I/O space registers. When this bit is 1, it enables the SATA controller to respond to PCI IO space access. |
| Memory Access Enable | 1 | 0b | This bit controls access to the memory space registers. When this bit is 1, it enables the SATA controller to respond to PCI memory space access |
| Bus Master Enable | 2 | 0b | Bus master function enable.<br>1 = Enable<br>0 = Disable. |
| Special Cycle Recognition Enable | 3 | 0b | Read Only. Hardwired to '0' |
| Memory Write and Invalidate Enable | 4 | 0b | Read Only. Hardwired to '0' |
| VGA Palette Snoop Enable | 5 | 0b | Read Only. Hard-wired to '0' indicating that the SATA host controller does not need to snoop VGA palette cycles. |
| PERR- Detection Enable | 6 | 0b | If set to 1, the IDE host controller asserts PERR- when it is the agent receiving data AND it detects a parity error. PERR- is not asserted if this bit is 0. |
| Wait Cycle Enable | 7 | 0b | Read Only.<br>Hard-wired to '0' to indicate that the SATA controller does not need to insert a wait state between the address and data on the AD lines. |
| SERR- Enable | 8 | 0b | If set to 1, and bit 6 is set, then the SATA controller asserts SERR- when it detects an address parity error. SERR- is not asserted if this bit is 0. |
| Fast Back-to-Back Enable | 9 | 0b | Read Only. Hard-wired to '0' to indicate that fast back to back is only allowed to the same agent. |
| Interrupt Disable | 10 | 0b | Complies with the PCI 2.3 specification. |
| Reserved | 15:11 | | Reserved. |

| Status - RW - 16 bits - [PCI_Reg:06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | | Reserved. |
| Interrupt Status | 3 | 0b | Interrupt status bit. Complies with the PCI 2.3 specification. |
| Capabilities List | 4 | 1b | Read Only. Hardwired to 1 to indicate that the Capabilities Pointer is located at 34h. |
| 66MHz Support | 5 | 1b | 66MHz capable. This feature is supported in the SATA controller. |
| Reserved | 6 | | Reserved. |
| Fast Back-to-Back Capable | 7 | 0b | Read Only. Hard-wired to '0' to indicate that it is fast back to back incapable. |
| Data Parity Error | 8 | 0b | Data Parity reported. Set to 1 if the SATA controller detects PERR- asserted while acting as the PCI master (whether PERR- was driven by the SATA controller or not.). Write '1' to clear this bit. |
| DEVSEL- Timing | 10:9 | 01b | Read only.<br>These bits indicate DEVSEL- timing when performing a positive decode. Since DEVSEL- is asserted to meet the medium timing, these bits are encoded as 01b. |
| Signaled Target Abort | 11 | 0b | Signaled Target Abort. This bit is set to 1, when the SATA controller signals Target Abort. Write '1' to clear this bit. |
| Received Target Abort | 12 | 0b | Received Target Abort. This bit is set to 1 when the SATA controller that generated the PCI cycle (SATA controller is the PCI master) is aborted by a PCI target. Write '1' to clear this bit. |
| Received Master Abort Status | 13 | 0b | Received Master Abort Status. Set to 1 when the SATA controller, acting as a PCI master, aborts a PCI bus memory cycle. Write '1' to clear this bit.. |
| SERR- Status | 14 | 0b | SERR- status. This bit is set to 1 when the SATA controller detects a PCI address parity error. Write '1'to clear this bit. |
| Detected Parity Error | 15 | 0b | Detected Parity Error. This bit is set to 1 when the SATA controller detects a parity error. Write '1' to clear this bit. |

| Revision ID/Class Code- R - 32 bits - [PCI_Reg:08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | These bits are hardwired to 00h to indicate the revision level of the chip design. |
| Operating Mode Selection | 15:8 | 8Fh | RW<br>Programmable I/F.<br>Bit [15] = Master IDE Device. Always 1.<br>Bits [14:12] = Reserved. Always read as 0's.<br>Bit [11] = Programmable indicator for Secondary. Always 1 to indicate that both modes are supported.<br>Bit [10] = Operating Mode for Secondary.<br>1 = Native PCI-mode.<br>0 = Compatibility Mode<br>Bit [9] = Programmable indicator for Primary. Always 1 to indicate that both modes are supported.<br>Bit [8] = Operating Mode for Primary.<br>1 = Native PCI-mode.<br>0 = Compatibility |
| Sub-Class Code | 23:16 | 01h | Sub-Class Code. 01h to indicate an IDE Controller. See Note. |
| Class Code | 31:24 | 01h | Class Code. These 8 bits are read only and wired to 01h to indicate a Mass-Storage Controller. |

| Revision ID/Class Code- R - 32 bits - [PCI_Reg:08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Note: This field is only writeable when PCI_Reg:40h[0] is set.<br>Sub-Class Code Program Interface Controller Type<br>01 8F IDE<br>06 01 AHCI<br>04 00 RAID | | | |

| Cache Line Size - RW - 8 bits - [PCI_Reg:0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | | Reserved. |
| Cache Line Size Register | 7:4 | 0h | If the value is 1, then the cache line size is 16 DW (64 byte). |

| Master Latency Timer - RW - 8 bits - [PCI_Reg:0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | | Reserved. |
| Master Latency Timer | 7:3 | 00h | Master Latency Timer. This number, in units of clocks, represents the guaranteed time slice allowed to the IDE host controller for burst transactions. |

| Header Type - R - 8 bits - [PCI_Reg:0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 00h | Header Type. Since the IDE host controller is a single-function device, this register contains a value of 00h. |

| BIST Mode Type - RW - 8 bits - [PCI_Reg:0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Completion Code | 3:0 | 0h | Read Only.<br>Indicates the completion code status of BIST. A non-zero value indicates a failure. |
| Reserved | 5:4 | | Reserved. |
| Start BIST | 6 | 0b | Since bit [7] is 0, program this bit take no effect. |
| BIST Capable | 7 | 0b | Read Only. Hard-wired to '0' indicating that there is no HBA related BIST function. |

| Base Address 0 - RW - 32 bits - [PCI_Reg:10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 2:1 | | Reserved. |
| Primary IDE CS0 Base Address | 31:3 | 0000_0000h | Base Address for Primary IDE Bus CS0. This register is used for native mode only. Base Address 0 is not used in compatibility mode. |

| Base Address 1 - RW - 32 bits - [PCI_Reg:14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 1 | | Reserved. |
| Primary IDE CS1 Base Address | 31:2 | 0000_0000h | Base Address for Primary IDE Bus CS1. This register is used for native mode only. Base Address 1 is not used in compatibility mode. |

| Base Address 2 - RW - 32 bits - [PCI_Reg:18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 2:1 | | Reserved. |
| Secondary IDE CS0 Base Address | 31:3 | 0000_ 0000h | Base Address for Secondary IDE Bus CS0. This register is used for native mode only. Base Address 2 is not used in compatibility mode. |

| Base Address 3 - RW - 32 bits - [PCI_Reg:1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 1 | | Reserved. |
| Secondary IDE CS1 Base Address | 31:2 | 0000_ 0000h | Base Address for Secondary IDE Bus CS1. This register is used for native mode only. Base Address 3 is not used in compatibility mode. |

| Bus Master Interface Base Address - RW - 32 bits - [PCI_Reg:20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 3:1 | | Reserved. |
| Bus Master Interface Register Base Address | 31:4 | 0000_ 000h | Base Address for the Bus Master interface registers, and corresponds to AD[15:4]. |

| AHCI Base Address - RW - 32 bits - [PCI_Reg:24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 0b | This bit is wired to 0 to indicate a request for register memory space. |
| Reserved | 9:1 | | Reserved. |
| AHCI Base Address | 31:10 | 000000h | Base address of register memory space. This represents a memory space for support of 4 ports. |

| Subsystem ID and Subsystem Vendor ID - RW - 32 bits - [PCI_Reg:2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Subsystem Vendor ID. This can only be written once by the software. |
| Subsystem ID | 31:16 | 0000h | Subsystem ID. This can only be written once by the software. |

| Capabilities Pointer - R - 8 bits - [PCI_Reg:34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 60h | The first pointer of the Capability block |

| Interrupt Line - RW - 8 bits - [PCI_Reg:3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | Identifies which input on the interrupt controller the function's PCI interrupt request pin (as specified in its Interrupt Pin register) is routed to. |

| Interrupt Pin - R - 8 bits - [PCI_Reg:3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 01h | Hard-wired to 01h. |

| Min_gnt - R - 8 bits - [PCI_Reg:3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minimum Grant | 7:0 | 00h | This register specifies the desired settings for how long of a burst the SATA controller needs. The value specifies a period of time in units of ¼ microseconds.<br>Hard-wired to 0's and always read as 0's. |

| Max_latency - R - 8 bits - [PCI_Reg:3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Maximum Latency | 7:0 | 00h | This register specifies the Maximum Latency time required before the SATA controller as a bus-master can start an accesses<br>Hard-wired to 0's and always read as 0's. |

| Misc Control - RW - 32 bits - [PCI_Reg:40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subclass code write Enable | 0 | 0b | Once set, Program Interface register (PCI_Reg:09h), subclass code register (PCI_Reg:0Ah) and Multiple Message Capable bits (PCI_Reg50h[19:17]) can be programmable. |
| Disable Dynamic Sata Memory Power Saving | 1 | 0b | When clear, dynamic power saving function for SATA internal memory macros will be performed to reduce power consumption. |
| Enable dynamic Sata Core Power Saving | 2 | 0b | When set, dynamic power saving function for SATA core clock will be performed during partial/slumber mode to reduce power consumption. |
| Reserved | 3 | | Reserved. |
| Disable Speed up XP Boot | 4 | 0b | When clear, it fastens XP boot up in IDE mode. However, this bit needs to be set, when enable SATA partial/slumber power function is in IDE mode.<br>When set, the SATA partial/slumber power function can be enabled in IDE mode, but the BIOS IO trap is needed to speed up XP boot-up in IDE mode.<br>Please refer to BAR5 + offset 12C/1Ac/22C/2AC[11:8] for the SATA partial/slumber modes that are allowed. |
| Reserved | 5 | 0b | Reserved |
| Reserved | 15:6 | | Reserved. |
| Disable port0 | 16 | 0b | When set, port0 is disabled and port0 clock is shut down. |
| Disable port1 | 17 | 0b | When set, port1 is disabled and port1 clock is shut down. |
| Disable port2 | 18 | 0b | When set, port2 is disabled and port2 clock is shut down. |
| Disable port3 | 19 | 0b | When set, port3 is disabled and port3 clock is shut down. |
| Reserved | 31:20 | | Reserved. |

| Watch Dog Control And Status - RW - 16 bits - [PCI_Reg:44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Watchdog Enable | 0 | 0b | Set this bit to enable the watchdog counter for all the PCI down stream transaction. |
| Watchdog Timeout Status | 1 | 0b | Watchdog Counter Timeout Status bit. This bit indicates that the watchdog counter has expired for PCI down stream transaction and the transaction got aborted due to counter has expired.<br>Software writes 1 to clear the status. |
| Reserved | 15:2 | | Reserved. |

| Watch Dog Counter - RW - 16 bits - [PCI_Reg:46h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Watchdog Counter | 7:0 | 80h | Specifies the timeout retry count for PCI down stream retries. |
| Reserved | 15:8 | | Reserved. |

| MSI Control - RW- 32 bits - [PCI_Reg:50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 05h | Read-Only.<br>Capability ID. It indicates that this is and MSI capability ID. |
| Capability Next Pointer | 15:8 | 70h | Read-Only.<br>Next Pointer (hard wired to 70h, points to Index Data pair capability |
| Message Signaled Interrupt Enable | 16 | 0b | MSI Enable. |
| Multiple Message Capable | 19:17 | 010b | Multiple Message Capable. |
| Multiple Message Enable | 22:20 | 000b | Multiple Message Enable. |
| MSI 64-bit Address | 23 | 1b | Read-Only<br>Support 64-bit address. |
| Reserved | 31:24 | | Reserved. |

| MSI Address - RW- 32 bits - [PCI_Reg:54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | | Reserved. |
| MSI Address | 31:2 | 0000_0000h | Lower 32 bits of the system specified message address always DW aligned. |

| MSI Upper Address - RW- 32 bits - [PCI_Reg:58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Upper Address | 31:0 | 0000_0000h | Upper 32 bits of the system specified message address. This register is optional and only implemented if MC.C64=1. |

| MSI Data - RW- 16 bits - [PCI_Reg:5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Data | 15:0 | 0000h | MSI Data |

| Power Management Capability ID - R- 16 bits - [PCI_Reg:60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 01h | Capability ID. Indicates this is power management capability ID. |
| Capability Next Pointer | 15:8 | 50h | Next Pointer. |

| Power Management Capability - R- 16 bits - [PCI_Reg:62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Version | 2:0 | 010b | Indicates support for Revision 1.1 of the *PCI Power Management Specification.* |
| PME Clock | 3 | 0b | Indicates that PCI clock is not required to generate PME#. |
| Reserved | 4 | | Reserved |
| Device Specific Initialization | 5 | 1b | Indicates whether device-specific initialization is required. |
| Aux_Current | 8:6 | 000b | Reports the maximum Suspend well current required when in the $D3_{COLD}$ state. Hard wired to 000b. |
| D1_Support | 9 | 0b | The D1 state is not supported. |
| D2_Support | 10 | 0b | The D2 state is not supported. |
| PME_Support | 15:11 | 00h | Read-Only. |

| PCI Power Management Control And Status - RW- 16 bits - [PCI_Reg:64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Power State | 1:0 | 00b | This field is used both to determine the current power state of the HBA and to set a new power state. The values are:<br>00 – D0 state<br>11 – $D3_{HOT}$ state<br>The D1 and D2 states are not supported. When in the $D3_{HOT}$ state, the configuration space is available, but the register memory spaces are not. Additionally, interrupts are blocked. |
| Reserved | 7:2 | | Reserved |
| PME Enable | 8 | 0b | Read-only. Hard-wired to '0' indicates PME disable |
| Reserved | 14:9 | | Reserved. |
| PME Status | 15 | 0b | Read-only. Hard-wired to '0' as PME disable |

| Serial ATA Capability Register 0 - R- 32 bits - [PCI_Reg:70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 12h | Capability ID. Indicates this is a Serial ATA Capability ID. |
| Capability Next Pointer | 15:8 | 00h | Next Pointer, end of the list. |
| Minor Revision | 19:16 | 0h | Minor revision number of the SATA Capability Pointer implemented. |
| Major Revision | 23:20 | 1h | Major revision number of the SATA Capability Pointer implemented. |
| Reserved | 31:24 | | Reserved |

| Serial ATA Capability Register 1 - R- 32 bits - [PCI_Reg:74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BAR Location | 3:0 | 1111b | Value 1111b indicates that the Index-Data Pair is implemented in Dwords directly following SATACR1 in the PCI configuration space. |
| BAR Offset | 23:4 | 00000h | Indicates the offset into the BAR where the Index-Data Pair are located in Dword granularity. Since the BAR location is setting at 1111b, this field is not used anymore. |
| Reserved | 31:24 | | Reserved |

| IDP Index Register - RW- 32 bits - [PCI_Reg:78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | | Reserved |
| IDP Index | 9:2 | 00h | This register selects the Dword offset of the memory mapped AHCI register to be accessed. The IDP Index should be sized such that it can access the entire ABAR register space for the particular implementation. See Note. |
| Reserved | 31:10 | | Reserved |
| Note: ABAR is AHCI memory map registers located at AHCI base address (BAR5) space. | | | |

| IDP Data Register - RW- 32 bits - [PCI_Reg:7Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDP Data | 31:0 | | This register is a "window" through which data is read or written to the memory mapped register pointed to by the IDP Index register. Note that a physical register is not actually implemented as the data is actually stored in the memory mapped registers.<br>Since this is not a physical register, the "default" value is the same as the default value of the register pointed to by IDP Index. |
| All register accesses to IDP Data are Dword granularity | | | |

| PHY Port0 Control - RW- 32 bits - [PCI_Reg:88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port0 PHY | 23:0 | B40014h | PHY port0 fine-tune register. |
| TX main driver swing | 4:0 | 10100b | Port0 Tx driving swing[4:0] is valid for SATA 1.5G. It sets the TX main driver swing. The user can program the optimum value for each SATA port. |
| | | | Bit 4 Bit 3 Bit 2 Bit 1 Bit0 Nominal Output |
| | | | 1 0 0 0 0 400mv |
| | | | 1 0 0 1 0 450mv |
| | | | 1 0 1 0 0 500mv |
| | | | 1 0 1 1 0 550mv |
| | | | 1 1 0 0 0 600mv |
| | | | 1 1 0 1 0 650mv |
| | | | 1 1 1 0 0 700mv |
| | | | 1 1 1 1 0 750mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis driver swing | 7:5 | 000b | Port0 Tx driving swing[7:5] is valid for both SATA 3G and 1.5G. It sets the TX pre-emphasis driver strength. The user can program the optimum pre-emphasis value for each SATA port if TX pre-emphasis enable bit is turned on. |
| | | | Bit 7 Bit 6 Bit 5 pre-emphasis amount |
| | | | 0 0 0 0mv |
| | | | 0 0 1 25mv |
| | | | 0 1 0 50mv |
| | | | 0 1 1 75mv |
| | | | 1 0 0 100mv |
| | | | 1 0 1 125mv |
| | | | 1 1 0 150mv |
| | | | 1 1 1 175mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis enable | 13 | 0b | Turns on TX pre-emphasis output<br>1: Enable pre-emphasis<br>0: Disable pre-emphasis |
| Reserved | 31:24 | | Reserved. |

| PHY Port1 Control - RW- 32 bits - [PCI_Reg:8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port1 PHY | 23:0 | B40014h | PHY port1 fine-tune register. |
| TX main swing | 4:0 | 10100b | Port1 Tx driving swing[4:0] is valid at SATA 1.5G. It sets the TX main driver swing. The user can program the optimum value for each SATA port. |
| | | | Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Nominal Output |
| | | | 1 0 0 0 0 400mv |
| | | | 1 0 0 1 0 450mv |
| | | | 1 0 1 0 0 500mv |
| | | | 1 0 1 1 0 550mv |
| | | | 1 1 0 0 0 600mv |
| | | | 1 1 0 1 0 650mv |
| | | | 1 1 1 0 0 700mv |
| | | | 1 1 1 1 0 750mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |

| PHY Port1 Control - RW- 32 bits - [PCI_Reg:8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TX pre-emphasis driver swing | 7:5 | 000b | Port1 Tx driving swing[7:5] is valid for both SATA 3G and 1.5G. It sets the TX pre-emphasis driver strength. The user can program the optimum pre-emphasis value for each SATA port if TX pre-emphasis enable bit turned on. |
| | | | Bit 7 Bit 6 Bit 5 pre-emphasis amount |
| | | | 0 0 0 0mv |
| | | | 0 0 1 25mv |
| | | | 0 1 0 50mv |
| | | | 0 1 1 75mv |
| | | | 1 0 0 100mv |
| | | | 1 0 1 125mv |
| | | | 1 1 0 150mv |
| | | | 1 1 1 175mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis enable | 13 | 0b | Turns on port1 TX pre-emphasis output.<br>1: Enable pre-emphasis<br>0: Disable pre-emphasis |

| PHY Port2 Control - RW- 32 bits - [PCI_Reg:90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port2 PHY | 23:0 | B40014h | PHY port2 fine-tune register. |
| TX main swing | 4:0 | 10100b | Port2 Tx driving swing[4:0] is valid at SATA 1.5G. |
| | | | Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Nominal Output |
| | | | 1 0 0 0 0 400mv |
| | | | 1 0 0 1 0 450mv |
| | | | 1 0 1 0 0 500mv |
| | | | 1 0 1 1 0 550mv |
| | | | 1 1 0 0 0 600mv |
| | | | 1 1 0 1 0 650mv |
| | | | 1 1 1 0 0 700mv |
| | | | 1 1 1 1 0 750mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis driver swing | 7:5 | 000b | Port2 Tx driving swing[7:5] is valid for both SATA 3G and 1.5G. It sets the TX pre-emphasis driver strength. The user can program the optimum pre-emphasis value for each SATA port if TX pre-emphasis enable bit turned on. |
| | | | Bit 7 Bit 6 Bit 5 pre-emphasis amount |
| | | | 0 0 0 0mv |
| | | | 0 0 1 25mv |
| | | | 0 1 0 50mv |
| | | | 0 1 1 75mv |
| | | | 1 0 0 100mv |
| | | | 1 0 1 125mv |
| | | | 1 1 0 150mv |
| | | | 1 1 1 175mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis enable | 13 | 0b | Turns on port2 TX pre-emphasis output<br>1: Enable pre-emphasis<br>0: Disable pre-emphasis |

| PHY Port3 Control - RW- 32 bits - [PCI_Reg:94h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port3 PHY | 23:0 | B40014h | PHY port3 fine-tune register. |
| TX main swing | 4:0 | 10100b | Port3 Tx driving swing[4:0] is valid at SATA 1.5G. |
| | | | Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Nominal Output |
| | | | 1 0 0 0 0 400mv |
| | | | 1 0 0 1 0 450mv |
| | | | 1 0 1 0 0 500mv |
| | | | 1 0 1 1 0 550mv |
| | | | 1 1 0 0 0 600mv |
| | | | 1 1 0 1 0 650mv |
| | | | 1 1 1 0 0 700mv |
| | | | 1 1 1 1 0 750mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis driver swing | 7:5 | 000b | Port3 Tx driving swing[7:5] is valid for both SATA 3G and 1.5G. It sets the TX pre-emphasis driver strength. The user can program the optimum pre-emphasis value for each SATA port if TX pre-emphasis enable bit turned on. |
| | | | Bit 7 Bit 6 Bit 5 pre-emphasis amount |
| | | | 0 0 0 0mv |
| | | | 0 0 1 25mv |
| | | | 0 1 0 50mv |
| | | | 0 1 1 75mv |
| | | | 1 0 0 100mv |
| | | | 1 0 1 125mv |
| | | | 1 1 0 150mv |
| | | | 1 1 1 175mv |
| | | | Note: This applies to all the ASIC Revisions A11 and above. |
| TX pre-emphasis enable | 13 | 0b | Turns on port3 TX pre-emphasis output<br>1: Enable pre-emphasis<br>0: Disable pre-emphasis |

| BIST Pattern Count - RW - 32 bits - [PCI_Reg:C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST Pattern Count | 31:0 | 0000_2000h | This count specifies how many Octal WORD pattern need to be checked before BIST Done bit be set. This count value is used fro all the 4 ports. 400h default value would be used for tester, which means 32K DWORD pattern would be compared for BIST test. Value of “0000_0000”h means the maximum patterns (16,000, 000, 000) checked. |

| PCI Target Control TimeOut Counter - RW – 8 bits - [PCI_Reg:C4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCI Target Control TimeOut Count | 7:0 | 80h | This register is used for programming the PCI Target Control TimeOut Count used to clear any stale target commands to the hosts controller. Granularity is 15.5us (Count * 15.5 us) The counter will be disabled if the count is programmed to 0x0. |

### *2.1.2 BAR0/BAR2/BAR1/BAR3 Registers (SATA I/O Register for IDE mode)*

BAR0/BAR2 uses 8 bytes of I/O space. BAR0 is used for Primary channel and BAR2 is used for Secondary channel during IDE native mode. BAR1/BAR3 uses 2 bytes of I/O space. BAR1 is used for Primary channel and BAR3 is used for Secondary channel during IDE native mode.

| Address (hex) | | | Name and Function | |
|---|---|---|---|---|
| **Compatibility Mode** | | **Native Mode (Offset)** | **Read Function** | **Write Function** |
| **IDE Command Block Registers** | | | | |
| **Primary** | **Secondary** | **BAR0/BAR2** | | |
| 1F0 | 170 | (Primary or Secondary) Base Address 0 + 0 | Data (16 bit) | Data (16 bit) |
| 1F1 | 171 | (Primary or Secondary) Base Address 0 + 1 | Error register | Features register |
| 1F2 | 172 | (Primary or Secondary) Base Address 0 + 2 | Sector Count | Sector Count |
| 1F3 | 173 | (Primary or Secondary) Base Address + 3 | Sector Number | Sector Number |
| 1F4 | 174 | (Primary or Secondary) Base Address + 4 | Cylinder Low | Cylinder Low |
| 1F5 | 175 | (Primary or Secondary) Base Address + 5 | Cylinder High | Cylinder High |
| 1F6 | 176 | (Primary or Secondary) Base Address + 6 | Drive/Head | Drive/Head |
| 1F7 | 177 | (Primary or Secondary) Base Address + 7 | Status | Command |
| **IDE Control Block Registers** | | | | |
| **Primary** | **Secondary** | **BAR1/BAR3** | | |
| 3F6 | 376 | (Primary or Secondary) Base Address + 2 | Alternate Status | Device Control |

### *2.1.3 BAR4 Registers (SATA I/O Register for IDE mode)*

BAR4 uses 16 bytes of I/O space. The Bus-master interface base address register (BAR4) defines the base address of the IO spare.

| Register Name | Offset Address [Primary/Secondary] |
|---|---|
| Bus-master IDE Command | 00h/08h |
| Bus-master IDE Status | 02h/0Ah |
| Descriptor Table Pointer | 04h/0Ch |

| **Bus-master IDE Command - RW- 8 bits - [IO_Reg: BAR4 + 00/08h]** | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Bus Master IDE Start/Stop | 0 | 0b | Bus Master IDE Start (1)/Stop (0).<br>This bit will not be reset by interrupt from IDE device. This must be reset by soft ware (device driver). |
| Reserved | 2:1 | | Reserved. |
| Bus Master Read/Write | 3 | 0b | Bus Master IDE r/w (direction) control<br>0 = Memory -> IDE<br>1 = IDE -> Memory<br>This bit should not change during Bus Master transfer cycle, even if terminated by Bus Master IDE stop. |
| Reserved | 7:4 | | Reserved. |

| Bus-master IDE Status - RW- 8 bits - [IO_Reg: BAR4 + 02/0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Bus Master Active | 0 | 0b | Bus Master IDE active. This bit is set to 1 when bit 0 in the Bus Master IDE command address register is set to 1. The IDE host controller sets this bit to 0 when the last transfer for a region is performed. This bit is also set to 0 when bit 0 of the Bus Master IDE command register is set to 0. |
| Bus Master DMA Error | 1 | 0b | IDE DMA error. This bit is set when the IDE host controller encounters a target abort, master abort, or Parity error while transferring data on the PCI bus. Write '1' clears this bit |
| IDE Interrupt | 2 | 0b | IDE Interrupt. Indicates when an IDE device has asserted its interrupt line. IRQ14 is used for the Primary channel and IRQ15 is used for the secondary channel. If the interrupt status bit is set to 0, by writing a 1 to this bit while the interrupt line is still at the active level, this bit remains 0 until another assertion edge is detected on the interrupt line. |
| Reserved | 4:3 | | Reserved. |
| Master Device DMA Capable | 5 | 0b | Device 0 (Master) DMA capable. |
| Slave Device DMA Capable | 6 | 0b | Device 1 (Slave) DMA capable. |
| Simplex Only | 7 | 0b | Read Only<br>Simplex only. This bit is hard-wired as 0. |

| Descriptor Table Pointer - RW- 32 bits - [IO_Reg: BAR4 + 04/0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 0h | Reserved. Always read as 0's. |
| Descriptor Table Base Address | 31:2 | 0000_0000h | Base Address of Descriptor Table. These bits correspond to Address [31-02]. |

### *2.1.4 BAR5 Registers*

These are the AHCI memory map registers. The base address is defined through the ABAR (BAR5) register.

| Register Name | Offset Address |
|---|---|
| Generic Host Control | 00h-23h |
| *Reserved* | 24h-9Fh |
| Vendor Specific registers | A0h-FFh |
| Port 0 port control registers | 100h-17Fh |
| Port 1 port control registers | 180h-1FFh |
| Port 2 port control registers | 200h-27Fh |
| Port 3 port control registers | 280h-2FFh |

#### *2.1.4.1 Generic Host Control*

The following registers apply to the entire HBA.

| Register Name | Offset Address |
|---|---|
| Host Capabilities(CAP) | 00h-03h |
| Global Host Control(GHC) | 04h-07h |
| Interrupt Status(IS) | 08h-0Bh |
| Ports Implemented(PI) | 0Ch-0Fh |
| Version(VS) | 10h-13h |
| Command Completion Coalescing Control(CCC_CTL) | 14h-17h |
| Command Completion Coalescing Ports(CCC_PORTS) | 18h-1Bh |
| Enclosure Management Location(EM_LOC) | 1Ch-1Fh |
| Enclosure Management Control(EM_CTL) | 20h-23h |

| HBA Capabilities – R - 32bits [Mem_reg: ABAR + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Number of Ports(NP) | 4:0 | 00011b | 0's based value indicating the maximum number of ports supported by the HBA silicon. A maximum of 32 ports can be supported. A value of '0h', indicating one port, is the minimum requirement. Note that the number of ports indicated in this field may be more than the number of ports indicated in the GHC.PI register. |
| Supports External SATA (SXS) | 5 | 0b | When set to '1', indicates that the HBA has one or more Serial ATA ports that has a signal only connector that is externally accessible. If this bit is set to '1', software may refer to the PxCMD.ESP bit to determine whether a specific port has its signal connector externally accessible as a signal only connector (i.e. power is not part of that connector). When the bit is cleared to '0', indicates that the HBA has no Serial ATA ports that have a signal only connector externally accessible. |
| Enclosure Management Supported (EMS) | 6 | 0b | When set to '1', indicates that the HBA supports enclosure management. When enclosure management is supported, the HBA has implemented the EM_LOC and EM_CTL global HBA registers. When cleared to '0', indicates that the HBA does not support enclosure management and the EM_LOC and EM_CTL global HBA registers are not implemented. |
| Command Completion Coalescing Supported (CCCS) | 7 | 1b | When set to '1', indicates that the HBA supports command completion coalescing. When command completion coalescing is supported, the HBA has implemented the CCC_CTL and the CCC_PORTS global HBA registers. When cleared to '0', indicates that the HBA does not support command completion coalescing and the CCC_CTL and CCC_PORTS global HBA registers are not implemented. |
| Number of Command Slots (NCS) | 12:8 | 11111b | 0's based value indicating the number of command slots per port supported by this HBA. A minimum of 1 and maximum of 32 slots per port can be supported. The same number of command slots is available on each implemented port. |
| Partial State Capable (PSC) | 13 | 1b | Indicates whether the HBA can support transitions to the Partial state. When cleared to '0', software must not allow the HBA to initiate transitions to the Partial state via aggressive link power management nor the PxCMD.ICC field in each port, and the PxSCTL.IPM field in each port must be programmed to disallow device initiated Partial requests. When set to '1', HBA and device initiated Partial requests can be supported. |
| Slumber State Capable (SSC) | 14 | 1b | Indicates whether the HBA can support transitions to the Slumber state. When cleared to '0', software must not allow the HBA to initiate transitions to the Slumber state via aggressive link power management nor the PxCMD.ICC field in each port, and the PxSCTL.IPM field in each port must be programmed to disallow device initiated Slumber requests. When set to '1', HBA and device initiated Slumber requests can be supported. |
| PIO Multiple DRQ Block (PMD) | 15 | 1b | If set to '1', the HBA supports multiple DRQ block data transfers for the PIO command protocol. If cleared to '0' the HBA only supports single DRQ block data transfers for the PIO command protocol. |
| FIS-based Switching Supported (FBSS) | 16 | 0b | When set to '1', indicates that the HBA supports Port Multiplier FIS-based switching. When cleared to '0', indicates that the HBA does not support FIS-based switching. AHCI 1.0 and 1.1 HBAs shall have this bit cleared to '0'. |

<table>
<tr><th colspan="4">HBA Capabilities – R - 32bits [Mem_reg: ABAR + 00h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>Supports Port Multiplier (SPM)</td><td>17</td><td>1b</td><td>Indicates whether the HBA can support a Port Multiplier. When set, a Port Multiplier using command-based switching is supported. When cleared to ‘0’, a Port Multiplier is not supported, and a Port Multiplier may not be attached to this HBA.</td></tr>
<tr><td>Supports AHCI mode only (SAM)</td><td>18</td><td>0b</td><td>The SATA controller may optionally support AHCI access mechanisms only. A value of '0' indicates that in addition to the native AHCI mechanism (via ABAR), the SATA controller implements a legacy, task-file based register interface such as SFF-8038i. A value of '1' indicates that the SATA controller does not implement a legacy, task-file based register interface.</td></tr>
<tr><td>Supports Non-Zero DMA Offsets (SNZO)</td><td>19</td><td>0b</td><td>When set to ‘1’, indicates that the HBA can support non-zero DMA offsets for DMA Setup FISes. This bit is reserved for future AHCI enhancements. AHCI 1.0 and 1.1 HBAs shall have this bit cleared to ‘0’.</td></tr>
<tr><td>Interface Speed Support (ISS)</td><td>23:20</td><td>2h</td><td>Indicates the maximum speed the HBA can support on its ports. These encodings match the system software programmable PxSCTL.DET.SPD field. Values are:
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>0000</td><td>Reserved</td></tr>
<tr><td>0001</td><td>Gen 1 (1.5 Gbps)</td></tr>
<tr><td>0010</td><td>Gen 1 (1.5 Gbps) and Gen 2 (3 Gbps)</td></tr>
<tr><td>0011 - 1111</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Supports Command List Override (SCLO)</td><td>24</td><td>1b</td><td>When set to ‘1’, the HBA supports the PxCMD.CLO bit and its associated function. When cleared to ‘0’, the HBA is not capable of clearing the BSY and DRQ bits in the Status register in order to issue a software reset if these bits are still set from a previous operation.</td></tr>
<tr><td>Supports Activity LED (SAL)</td><td>25</td><td>1b</td><td>When set to ‘1’, the HBA supports a single activity indication output pin. This pin can be connected to an LED on the platform to indicate device activity on any drive. When cleared to ‘0’, this function is not supported.</td></tr>
<tr><td>Supports Aggressive Link Power Management (SALP)</td><td>26</td><td>1b</td><td>When set to ‘1’, the HBA can support auto-generating link requests to the Partial or Slumber states when there are no commands to process. When cleared to ‘0’, this function is not supported and software shall treat the PxCMD.ALPE and PxCMD.ASP bits as reserved.</td></tr>
<tr><td>Supports Staggered Spin-up (SSS)</td><td>27</td><td>0b</td><td>When set to ‘1’, the HBA supports staggered spin-up on its ports, for use in balancing power spikes. When cleared to ‘0’, this function is not supported. This value is loaded by the BIOS prior to OS initialization.</td></tr>
<tr><td>Supports Mechanical Presence Switch (SMPS)</td><td>28</td><td>1b</td><td>When set to ‘1’, the HBA supports mechanical presence switches on its ports for use in hot plug operations. When cleared to ‘0’, this function is not supported. This value is loaded by the BIOS prior to OS initialization.</td></tr>
<tr><td>Supports SNotification Register (SSNTF)</td><td>29</td><td>1b</td><td>When set to ‘1’, the HBA supports the PxSNTF (SNotification) register and its associated functionality. When cleared to ‘0’, the HBA does not support the PxSNTF (SNotification) register and its associated functionality.</td></tr>
</table>

| HBA Capabilities – R - 32bits [Mem_reg: ABAR + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Supports Native Command Queuing (SNCQ) | 30 | 1b | Indicates whether the HBA supports Serial ATA native command queuing. If set to '1', an HBA shall handle DMA Setup FISes natively, and shall handle the auto-activate optimization through that FIS. If cleared to '0', native command queuing is not supported and software should not issue any native command queuing commands. |
| Supports 64-bit Addressing (S64A) | 31 | 1b | Indicates whether the HBA can access 64-bit data structures. When set to '1', the HBA shall make the 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry read/write. When cleared to '0', these are read-only and treated as '0' by the HBA. |

| Global HBA Control – RW - 32bits [Mem_reg: ABAR + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| HBA Reset (HR) | 0 | 0b | When set by SW, this bit causes an internal reset of the HBA. All state machines that relate to data transfers and queuing shall return to an idle condition, and all ports shall be re-initialized via COMRESET (if staggered spin-up is not supported). If staggered spin-up is supported, then it is the responsibility of software to spin-up each port after the reset has completed.<br>When the HBA has performed the reset action, it shall reset this bit to '0'. A software write of '0' shall have no effect. For a description on which bits are reset when this bit is set. |
| Interrupt Enable (IE) | 1 | 0b | This global bit enables interrupts from the HBA. When cleared (reset default), all interrupt sources from all ports are disabled. When set, interrupts are enabled. |
| MSI Revert to Single Message (MRSM) | 2 | 0b | Read Only<br>When set to '1' by hardware, indicates that the HBA requested more than one MSI vector but has reverted to using the first vector only. When this bit is cleared to '0', the HBA has not reverted to single MSI mode (i.e. hardware is already in single MSI mode, software has allocated the number of messages requested, or hardware is sharing interrupt vectors if MC.MME < MC.MMC).<br>The HBA may revert to single MSI mode when the number of vectors allocated by the host is less than the number requested. This bit shall only be set to '1' when the following conditions hold:<br>• MC.MSIE = '1' (MSI is enabled)<br>• MC.MMC > 0 (multiple messages requested)<br>• MC.MME > 0 (more than one message allocated)<br>• MC.MME != MC.MMC (messages allocated not equal to number requested)<br>When this bit is set to '1', single MSI mode operation is in use and software is responsible for clearing bits in the IS register to clear interrupts.<br>This bit shall be cleared to '0' by hardware when any of the four conditions stated is false. This bit is also cleared to '0' when MC.MSIE = '1' and MC.MME = 0h. In this case, the hardware has been programmed to use single MSI mode, and is not "reverting" to that mode. |
| Reserved | 30:3 | | Reserved. |

| Global HBA Control – RW - 32bits [Mem_reg: ABAR + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AHCI Enable (AE) | 31 | 0b | When set, indicates that communication to the HBA shall be via AHCI mechanisms. This can be used by an HBA that supports both legacy mechanisms (such as SFF-8038i) and AHCI to know when the HBA is running under an AHCI driver.<br>When set, software shall only communicate with the HBA using AHCI. When cleared, software shall only communicate with the HBA using legacy mechanisms. When cleared FISes are not posted to memory and no commands are sent via AHCI mechanisms.<br>Software shall set this bit to '1' before accessing other AHCI registers. |

| Interrupt Status - RW -32 bits [Mem_reg: ABAR + 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pending Status (IPS) | 31:0 | 0000_0000h | If set, indicates that the corresponding port has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>The IPS[x] bit is only defined for ports that are implemented or for the command completion coalescing interrupt defined by CCC_CTL.INT. All other bits are reserved. |
| Write 1 to clear these status bits. | | | |

| Ports Implemented Register - R -32 bits [Mem_reg: ABAR + 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port Implemented (PI) | 31:0 | 0000000Fh | This register is bit significant. If a bit is set to '1', the corresponding port is available for software to use. If a bit is cleared to '0', the port is not available for software to use. The maximum number of bits set to '1' shall not exceed CAP.NP + 1, although the number of bits set in this register may be fewer than CAP.NP + 1. At least one bit shall be set to '1'. |

| AHCI Version- R – 32 bits [Mem_reg: ABAR + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minor Version Number (MNR) | 15:0 | 0100h | Indicates the minor version is "10". |
| Major Version Number (MJR) | 31:16 | 0001h | Indicates the major version is "1" |
| Version: V1.10 | | | |

| Command Completion Coalescing Control(CCC_CTL) - RW – 32 bits [Mem_reg: ABAR + 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CCC_CTL Enable | 0 | 0b | When cleared to '0', the command completion coalescing feature is disabled and no CCC interrupts are generated. When set to '1', the command completion coalescing feature is enabled and CCC interrupts may be generated based on timeout or command completion conditions. Software shall only change the contents of the TV and CC fields when EN is cleared to '0'. On transition of this bit from '0' to '1', any updated values for the TV and CC fields shall take effect. |
| Reserved | 2:1 | | Reserved |

| Command Completion Coalescing Control(CCC_CTL) - RW – 32 bits [Mem_reg: ABAR + 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CCC Interrupt (INT) | 7:3 | 1Fh | Read Only<br>Specifies the interrupt used by the CCC feature. This interrupt must be marked as unused in the Ports Implemented (PI) register by the corresponding bit being set to '0'. Thus, the CCC interrupt corresponds to the interrupt for an unimplemented port on the controller. When a CCC interrupt occurs, the IS.IPS[INT] bit shall be asserted to '1'. This field also specifies the interrupt vector used for MSI. |
| Command Completions (CC) | 15:8 | 01h | Specifies the number of command completions that are necessary to cause a CCC interrupt. The HBA has an internal command completion counter, hCccComplete. hCccComplete is incremented by one each time a selected port has a command completion. When hCccComplete is equal to the command completions value, a CCC interrupt is signaled. The internal command completion counter is reset to '0' on the assertion of each CCC interrupt. A value of '0' for this field shall disable CCC interrupts being generated based on the number of commands completed, i.e. CCC interrupts are only generated based on the timer in this case. |
| Timeout Value (TV) | 31:16 | 0001h | The timeout value is specified in 1 millisecond intervals. The timer accuracy shall be within 5%. hCccTimer is loaded with this timeout value. The hCccTimer is only decremented when commands are outstanding on selected ports. The HBA will signal a CCC interrupt when hCccTimer has decremented to '0'. The hCccTimer is reset to the timeout value on the assertion of each CCC interrupt. A timeout value of '0' is reserved. |

| Command Completion Coalescing Ports - RW – 32 bits [Mem_reg: ABAR + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Ports (PRT) | 31:0 | 00000000h | This register is bit significant. Each bit corresponds to a particular port, where bit 0 corresponds to port 0. If a bit is set to '1', the corresponding port is part of the command completion coalescing feature. If a bit is cleared to '0', the port is not part of the command completion coalescing feature. Bits set to '1' in this register must also have the corresponding bit set to '1' in the Ports Implemented register. An updated value for this field shall take effect within one timer increment (1 millisecond). |

#### *2.1.4.2 Port Registers (One Set Per Port)*

The algorithm for the software to determine the offset is as follows:

- Port offset = 100h + (PI Asserted Bit Position * 80h)

| Register Name | Offset Address |
|---|---|
| Port-N Command List Base Address(PNCLB) | 00h-03h + Port offset |
| Port-N Command List Base Address Upper 32-Bits(PNCLBU) | 04h-07h + Port offset |
| Port-N FIS Base Address(PNFB) | 08h-0Bh + Port offset |
| Port-N FIS Base Address Upper 32-Bits(PNFBU) | 0Ch-0Fh + Port offset |
| Port-N Interrupt Status(PNIS) | 10h-13h + Port offset |
| Port-N Interrupt Enable(PNIE) | 14h-17h + Port offset |
| Port-N Command and Status(PNCMD) | 18h-1Bh + Port offset |
| Reserved | 1Ch-1Fh + Port offset |
| Port-N Task File Data(PNTFD) | 20h-23h + Port offset |
| Port-N Signature(PNSIG) | 24h-27h + Port offset |
| Port-N Serial ATA Status (PNSSTS) | 28h-2Bh + Port offset |
| Port-N Serial ATA Control (PNSCTL) | 2Ch-2Fh + Port offset |
| Port-N Serial ATA Error (PNSERR) | 30h-33h + Port offset |
| Port-N Serial ATA Active (PNSACT) | 34h-37h + Port offset |
| Port-N Command Issue(PNCI) | 38h-3Bh + Port offset |
| Port-N SNotification (PNSNTF) | 3Ch-3Fh + Port offset |
| Reserved for FIS-based Switching Definition | 40h-43h + Port offset |
| Reserved | 44h-6Fh + Port offset |
| Port-N Vendor Specific(PNVS) | 70h-7Fh + Port offset |
| *N is the port number, 0 ~ 3 | |

| Port-N Command List Base Address -RW -32 bits [Mem_reg: ABAR + port offset + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 9:0 | | Reserved. |
| Command List Base Address (CLB) | 31:10 | 000000h | Indicates the 32-bit base physical address for the command list for this port. This base is used when fetching commands to execute. The structure pointed to by this address range is 1K-bytes in length. This address must be 1K-byte aligned as indicated by bits 09:00 being read only. |

| Port-N Command List Base Upper Address -RW - 32 bits [Mem_reg: ABAR + port offset + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Command List Base Address Upper (CLBU) | 31:0 | 00000000h | Indicates the upper 32-bits for the command list base physical address for this port. This base is used when fetching commands to execute.<br>This register shall be read only '0' for HBAs that do not support 64-bit addressing. |

| Port–N FIS Base Address -RW -32 bits [Mem_reg: ABAR + port offset + 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | | Reserved. |
| FIS Base Address (FB) | 31:8 | 000000h | Indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256-byte aligned as indicated by bits 07:00 being read only. |

| Port-N FIS Base Address Upper –RW – 32 bits [Mem_reg: ABAR + port offset + 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FIS Base Address Upper (FBU) | 31:0 | 0000_ 0000h | Indicates the upper 32-bits for the received FIS base physical address for this port.<br>This register shall be read only '0' for HBAs that do not support 64-bit addressing. |

| Port–N Interrupt Status - RW - 32 bits [Mem_reg: ABAR + port offset + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device to Host Register FIS Interrupt (DHRS) | 0 | 0b | A D2H Register FIS has been received with the 'I' bit set, and has been copied into system memory. |
| PIO Setup FIS Interrupt (PSS) | 1 | 0b | A PIO Setup FIS has been received with the 'I' bit set, it has been copied into system memory, and the data related to that FIS has been transferred. This bit shall be set even if the data transfer resulted in an error. |
| DMA Setup FIS Interrupt (DSS) | 2 | 0b | A DMA Setup FIS has been received with the 'I' bit set and has been copied into system memory. |
| Set Device Bits Interrupt (SDBS) | 3 | 0b | A Set Device Bits FIS has been received with the 'I' bit set and has been copied into system memory. |
| Unknown FIS Interrupt (UFS) | 4 | 0b | Read Only<br>When set to '1', indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to '0' by software clearing the PxSERR.DIAG.F bit to '0'. Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when that FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to '1' or the two bits may become out of sync. |
| Descriptor Processed (DPS) | 5 | 0b | A PRD with the 'I' bit set has transferred all of its data. |
| Port Connect Change Status (PCS) | 6 | 0b | Read Only<br>1=Change in *Current Connect Status*. 0=No change in *Current Connect Status*. This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared. |
| Device Mechanical Presence Status (DMPS) | 7 | 0b | When set, indicates that a mechanical presence switch attached to this port has been opened or closed, which may lead to a change in the connection state of the device. This bit is only valid if both CAP.SMPS and P0CMD.MPSP are set to '1'. |
| Reserved | 21:8 | | Reserved |
| PhyRdy Change Status (PRCS) | 22 | 0b | Read Only<br>When set to '1' indicates the internal PhyRdy signal changed state. This bit reflects the state of P0SERR.DIAG.N. To clear this bit, software must clear P0SERR.DIAG.N to '0'. |
| Incorrect Port Multiplier Status (IPMS) | 23 | 0b | Indicates that the HBA received a FIS from a device whose Port Multiplier field did not match what was expected. The IPMS bit may be set during enumeration of devices on a Port Multiplier due to the normal Port Multiplier enumeration process. It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| Overflow Status (OFS) | 24 | 0b | Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| Reserved | 25 | | Reserved |
| Interface Non-fatal Error Status (INFS) | 26 | 0b | Indicates that the HBA encountered an error on the Serial ATA interface but was able to continue operation. |
| Interface Fatal Error Status (IFS) | 27 | 0b | Indicates that the HBA encountered an error on the Serial ATA interface which caused the transfer to stop. |

| Port–N Interrupt Status - RW - 32 bits [Mem_reg: ABAR + port offset + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Host Bus Data Error Status (HBDS) | 28 | 0b | Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| Host Bus Fatal Error Status (HBFS) | 29 | 0b | Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, such an indication would be a target or master abort. |
| Task File Error Status (TFES) | 30 | 0b | This bit is set whenever the status register is updated by the device and the error bit (bit 0) is set. |
| Cold Port Detect Status (CPDS) | 31 | 0b | When set, a device status has changed as detected by the cold presence detect logic. This bit can either be set due to a non-connected port receiving a device, or a connected port having its device removed. This bit is only valid if the port supports cold presence detect as indicated by PxCMD.CPD set to '1'. |
| Write 1 to clear these status bits. | | | |

| Port-N Interrupt Enable - RW -32 bits [Mem_reg: ABAR + port offset + 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device to Host Register FIS Interrupt Enable (DHRE) | 0 | 0b | When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |
| PIO Setup FIS Interrupt Enable (PSE) | 1 | 0b | When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| DMA Setup FIS Interrupt Enable (DSE) | 2 | 0b | When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| Set Device Bits FIS Interrupt Enable (SDBE) | 3 | 0b | When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| Unknown FIS Interrupt Enable (UFE) | 4 | 0b | When set, GHC.IE is set, and P0IS.UFS is set to '1', the HBA shall generate an interrupt. |
| Descriptor Processed Interrupt Enable (DPE) | 5 | 0b | When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| Port Change Interrupt Enable (PCE) | 6 | 0b | When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| Device Mechanical Presence Enable (DMPE) | 7 | 0b | When set, and GHC.IE is set to '1', and P0IS.DMPS is set, the HBA shall generate an interrupt.<br>For systems that do not support a mechanical presence switch, this bit shall be a read-only '0'. |
| Reserved | 21:8 | | Reserved |
| PhyRdy Change Interrupt Enable (PRCE) | 22 | 0b | When set to '1', and GHC.IE is set to '1', and P0IS.PRCS is set to '1', the HBA shall generate an interrupt. |
| Incorrect Port Multiplier Enable (IPME) | 23 | 0b | When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. |
| Overflow Enable (OFE) | 24 | 0b | When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| Reserved | 25 | | Reserved |
| Interface Non-fatal Error Enable (INFE) | 26 | 0b | When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| Interface Fatal Error Enable (IFE) | 27 | 0b | When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| Host Bus Data Error Enable (HBDE) | 28 | 0b | When set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| Host Bus Fatal Error Enable (HBFE) | 29 | 0b | When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| Task File Error Enable (TFEE) | 30 | 0b | When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |

| Port-N Interrupt Enable - RW -32 bits [Mem_reg: ABAR + port offset + 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cold Presence Detect Enable (CPDE) | 31 | 0b | When set, GHC.IE is set, and P0S.CPDS is set, the HBA shall generate an interrupt.<br>For systems that do not support cold presence detect, this bit shall be a read-only '0'. |

| Port-N Command and Status - R - 32 bits [Mem_reg: ABAR + port offset + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Start (ST) | 0 | 0b | RW<br>When set, the HBA may process the command list. When cleared, the HBA may not process the command list. Whenever this bit is changed from a '0' to a '1', the HBA starts processing the command list at entry '0'. Whenever this bit is changed from a '1' to a '0', the PxCI register is cleared by the HBA upon the HBA putting the controller into an idle state. This bit shall only be set to '1' by software after PxCMD.FRE has been set to '1'. |
| Spin-Up Device (SUD) | 1 | 1b | This bit is read/write for HBAs that support staggered spin-up via CAP.SSS. This bit is read only '1' for HBAs that do not support staggered spin-up. On an edge detect from '0' to '1', the HBA shall start a COMRESET initialization sequence to the device. Clearing this bit to '0' does not cause any OOB signal to be sent on the interface. When this bit is cleared to '0' and PxSCTL.DET=0h, the HBA will enter listen mode. |
| Power On Device (POD) | 2 | 1b | This bit is read/write for HBAs that support cold presence detection on this port as indicated by PxCMD.CPD set to '1'. This bit is read only '1' for HBAs that do not support cold presence detect. When set, the HBA sets the state of a pin on the HBA to '1' so that it may be used to provide power to a cold-presence detectable port. |
| Command List Override (CLO) | 3 | 0b | RW<br>Setting this bit to '1' causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to '0'. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register. The HBA sets this bit to '0' when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to '0'. A write to this register with a value of '0' shall have no effect.<br>This bit shall only be set to '1' immediately prior to setting the PxCMD.ST bit to '1' from a previous value of '0'. Setting this bit to '1' at any other time is not supported and will result in indeterminate behavior. Software must wait for CLO to be cleared to '0' before setting PxCMD.ST to '1'. |
| FIS Receive Enable (FRE) | 4 | 0b | RW<br>When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB (and for 64-bit HBAs, PxFBU). When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area.<br>System software must not set this bit until PxFB (PxFBU) have been programmed with a valid pointer to the FIS receive area, and if software wishes to move the base, this bit must first be cleared, and software must wait for the FR bit in this register to be cleared |
| Reserved | 7:5 | | Reserved |

| Port-N Command and Status - R - 32 bits [Mem_reg: ABAR + port offset + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Current Command Slot (CCS) | 12:8 | 00h | This field is valid when P0CMD.ST is set to '1' and shall be set to the command slot value of the command that is currently being issued by the HBA. When P0CMD.ST transitions from '1' to '0', this field shall be reset to '0'. After P0CMD.ST transitions from '0' to '1', the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is P0CMD.CCS + 1. For example, after the HBA has issued its first command, if CCS = 0h and P0CI is set to 3h, the next command that will be issued is from command slot 1. |
| Mechanical Presence Switch State (MPSS) | 13 | 1b | The MPSS bit reports the state of a mechanical presence switch attached to this port. If CAP.SMPS is set to '1' and the mechanical presence switch is closed then this bit is cleared to '0'. If CAP.SMPS is set to '1' and the mechanical presence switch is open then this bit is set to '1'. If CAP.SMPS is set to '0' then this bit is cleared to '0'. Software should only use this bit if both CAP.SMPS and P0CMD.MPSP are set to '1'. |
| FIS Receive Running (FR) | 14 | 0b | When set, the FIS Receive DMA engine for the port is running. |
| Command List Running (CR) | 15 | 0b | When this bit is set, the command list DMA engine for the port is running. See the AHCI state machine in section. |
| Cold Presence State (CPS) | 16 | 0b | The CPS bit reports whether a device is currently detected on this port via cold presence detection. If CPS is set to '1', then the HBA detects via cold presence that a device is attached to this port. If CPS is cleared to '0' , then the HBA detects via cold presence that there is no device attached to this port. |
| Port Multiplier Attached (PMA) | 17 | 0b | RW<br>This bit is read/write for HBAs that support a Port Multiplier (CAP.SPM = '1'). This bit is read-only for HBAs that do not support a port Multiplier (CAP.SPM = '0'). When set to '1' by software, a Port Multiplier is attached to the HBA for this port. When cleared to '0' by software, a Port Multiplier is not attached to the HBA for this port. Software is responsible for detecting whether a Port Multiplier is present; hardware does not auto-detect the presence of a Port Multiplier. |
| Hot Plug Capable Port (HPCP) | 18 | 1b | When set to '1', indicates that this port's signal and power connectors are externally accessible via a joint signal and power connector for blindmate device hot plug. When cleared to '0', indicates that this port's signal and power connectors are not externally accessible via a joint signal and power connector. |
| Mechanical Presence Switch Attached to Port (MPSP) | 19 | 0b | If set to '1', the platform supports an mechanical presence switch attached to this port. If cleared to '0', the platform does not support a mechanical presence switch attached to this port. When this bit is set to '1', P0CMD.HPCP should also be set to '1'. |
| Cold Presence Detection (CPD) | 20 | 0b | If set to '1', the platform supports cold presence detection on this port. If cleared to '0', the platform does not support cold presence detection on this port. When this bit is set to '1', P0CMD.HPCP should also be set to '1'. |
| External SATA Port (ESP) | 21 | 0b | When set to '1', indicates that this port's signal connector is externally accessible on a signal only connector. When set to '1', CAP.SXS shall be set to '1'. When cleared to '0', indicates that this port's signal connector is not externally accessible on a signal only connector. ESP is mutually exclusive with the HPCP bit in this register. |
| Reserved | 23:22 | | Reserved |

| Port-N Command and Status - R - 32 bits [Mem_reg: ABAR + port offset + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device is ATAPI (ATAPI) | 24 | 0b | RW<br>When set to '1', the connected device is an ATAPI device. This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| Drive LED on ATAPI Enable (DLAE) | 25 | 0b | RW<br>When set to '1', the HBA shall drive the LED pin active for commands regardless of the state of P0CMD.ATAPI. When cleared, the HBA shall only drive the LED pin active for commands if P0CMD.ATAPI set to '0'. |
| Aggressive Link Power Management Enable (ALPE) | 26 | 0b | RW<br>When set to '1', the HBA shall aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit. Software shall only set this bit to '1' if CAP.SALP is set to '1'; if CAP.SALP is cleared to '0' software shall treat this bit as reserved. |
| Aggressive Slumber / Partial (ASP) | 27 | 0b | RW<br>When set to '1', and ALPE is set, the HBA shall aggressively enter the Slumber state when it clears the PxCI register and the PxSACT register is cleared or when it clears the PxSACT register and PxCI is cleared. When cleared, and ALPE is set, the HBA shall aggressively enter the Partial state when it clears the PxCI register and the PxSACT register is cleared or when it clears the PxSACT register and PxCI is cleared. If CAP.SALP is cleared to '0' software shall treat this bit as reserved. |

<table>
<tr><th colspan="4">Port-N Command and Status - R - 32 bits [Mem_reg: ABAR + port offset + 18h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>Interface Communication Control (ICC)</td><td>31:28</td><td>0h</td><td>RW<br>This field is used to control power management states of the interface. If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>Fh - 7h</td><td>Reserved</td></tr>
<tr><td>6h</td><td>Slumber: This shall cause the HBA to request a transition of the interface to the Slumber state. The SATA device may reject the request and the interface shall remain in its current state.</td></tr>
<tr><td>5h - 3h</td><td>Reserved</td></tr>
<tr><td>2h</td><td>Partial: This shall cause the HBA to request a transition of the interface to the Partial state. The SATA device may reject the request and the interface shall remain in its current state.</td></tr>
<tr><td>1h</td><td>Active: This shall cause the HBA to request a transition of the interface into the active state.</td></tr>
<tr><td>0h</td><td>No-Op / Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not yet have occurred.</td></tr>
</table>
When the system software writes a non-reserved value other than No-Op (0h), the HBA shall perform the action and update this field back to Idle (0h).<br>If software writes to this field to change the state to a state the link is already in (i.e. interface is in the active state and a request is made to go to the active state), the HBA shall take no action and return this field to Idle. If the interface is in a low power state and software wants to transition to a different low power state, software must first bring the link to active and then initiate the transition to the desired low power state.</td></tr>
</table>

<table>
<tr><th colspan="4">Port-N Task Fike Data – R – 32 bits [Mem_reg: ABAR + port offset + 20h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>Status (STS)</td><td>7:0</td><td>7Fh</td><td>Contains the latest copy of the task file status register. Fields of note in this register that affect AHCI hardware operation are:<br><br><table><tr><th>Bit</th><th>Field</th><th>Definition</th></tr><tr><td>7</td><td>BSY</td><td>Indicates the interface is busy</td></tr><tr><td>6:4</td><td>cs</td><td>Command specific</td></tr><tr><td>3</td><td>DRQ</td><td>Indicates a data transfer is requested</td></tr><tr><td>2:1</td><td>cs</td><td>Command specific</td></tr><tr><td>0</td><td>ERR</td><td>Indicates an error during the transfer.</td></tr></table></td></tr>
<tr><td>ERROR</td><td>15:8</td><td>00h</td><td>Contains the latest copy of the task file error register.</td></tr>
<tr><td>Reserved</td><td>31:16</td><td></td><td>Reserved</td></tr>
</table>

<table>
<tr><th colspan="4">Port-N Signature – R – 32 bits [Mem_reg: ABAR + port offset + 24h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>Signature (SIG)</td><td>31:0</td><td>FFFFFFFFh</td><td>Contains the signature received from a device on the first D2H Register FIS. The bit order is as follows:<br><br><table><tr><th>Bit</th><th>Field</th></tr><tr><td>31:24</td><td>LBA High Register</td></tr><tr><td>23:16</td><td>LBA Mid Register</td></tr><tr><td>15:08</td><td>LBA Low Register</td></tr><tr><td>07:00</td><td>Sector Count Register</td></tr></table></td></tr>
<tr><td colspan="4">It is updated once after a reset sequence.</td></tr>
</table>

| Port-N Serial ATA Status – R – 32 bits [Mem_reg: ABAR + port offset + 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device Detection (DET) | 3:0 | 0h | Indicates the interface device detection and Phy state.<br>0h No device detected and Phy communication not established<br>1h Device presence detected but Phy communication not established<br>3h Device presence detected and Phy communication established<br>4h Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode<br>All other values reserved. Read Only |
| Current Interface Speed (SPD) | 7:4 | 0h | Indicates the negotiated interface communication speed.<br>0h Device not present or communication not established<br>1h Generation 1 communication rate negotiated<br>2h Generation 2 communication rate negotiated<br>All other values reserved. Read Only |

| Port-N Serial ATA Status – R – 32 bits [Mem_reg: ABAR + port offset + 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interface Power Management (IPM) | 11:8 | 0h | Indicates the current interface state:<br>0h Device not present or communication not established<br>1h Interface in active state<br>2h Interface in Partial power management state<br>6h Interface in Slumber power management state<br>All other values reserved. Read Only |
| Reserved | 31:12 | | Reserved |

| Port-N Serial ATA Control – RW – 32 bits [Mem_reg: ABAR + port offset + 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device Detection Initialization (DET) | 3:0 | 0h | Controls the HBA's device detection and interface initialization.<br>0h No device detection or initialization action requested<br>1h Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. While this field is 1h, COMRESET is transmitted on the interface. Software should leave the DET field set to 1h for a minimum of 1 millisecond to ensure that a COMRESET is sent on the interface.<br>4h Disable the Serial ATA interface and put Phy in offline mode.<br>All other values reserved<br>This field may only be modified when P0CMD.ST is '0'. Changing this field while the P0CMD.ST bit is set to '1' results in undefined behavior. When P0CMD.ST is set to '1', this field should have a value of 0h.<br>Note: It is permissible to implement any of the Serial ATA defined behaviors for transmission of COMRESET when DET=1h. |
| Speed Allowed (SPD) | 7:4 | 0h | Indicates the highest allowable speed of the interface.<br>0h No speed negotiation restrictions<br>1h Limit speed negotiation to Generation 1 communication rate<br>2h Limit speed negotiation to a rate not greater than Generation 2 communication rate<br>All other values reserved |

| Port-N Serial ATA Control – RW – 32 bits [Mem_reg: ABAR + port offset + 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interface Power Management Transitions Allowed (IPM) | 11:8 | 0h | Indicates which power states the HBA is allowed to transition to. If an interface power management state is disabled, the HBA is not allowed to initiate that state and the HBA must $PMNAK_P$ any request from the device to enter that state.<br>0h No interface restrictions<br>1h Transitions to the Partial state disabled<br>2h Transitions to the Slumber state disabled<br>3h Transitions to both Partial and Slumber states disabled<br>All other values reserved |
| Select Power Management (SPM) | 15:12 | 0h | Read Only |
| Port Multiplier Port (PMP) | 19:16 | 0h | Read Only |
| Reserved | 31:20 | | Reserved |

<table>
<tr><th colspan="4">Port-N Serial ATA Error – RW – 32 bits [Mem_reg: ABAR + port offset + 30h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>ERROR</td><td>15:0</td><td>0000h</td><td>The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br><br>
<table>
<tr><td>15:12</td><td>Reserved</td></tr>
<tr><td>11</td><td>Internal Error (E): The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. The internal error may include a master or target abort when attempting to access system memory, an elasticity buffer overflow, a primitive mis-alignment, a synchronization FIFO overflow, and other internal error conditions. Typically when an internal error occurs, a non-fatal or fatal status bit in the PxIS register will also be set to give software guidance on the recovery mechanism required.</td></tr>
<tr><td>10</td><td>Protocol Error (P): A violation of the Serial ATA protocol was detected.</td></tr>
<tr><td>9</td><td>Persistent Communication or Data Integrity Error (C): A communication error that was not recovered occurred that is expected to be persistent. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.</td></tr>
<tr><td>8</td><td>Transient Data Integrity Error (T): A data integrity error occurred that was not recovered by the interface. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error or 10b8b decoding error.</td></tr>
<tr><td>7:2</td><td>Reserved</td></tr>
<tr><td>1</td><td>Recovered Communications Error (M): Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PhyNRdy signal between the Phy and Link layers.</td></tr>
<tr><td>0</td><td>Recovered Data Integrity Error (I): A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error or 10b8b decoding error.</td></tr>
</table>
Write 1 to clear these bits.</td></tr>
</table>

<table>
<tr><th colspan="4">Port-N Serial ATA Error – RW – 32 bits [Mem_reg: ABAR + port offset + 30h]</th></tr>
<tr><th>Field Name</th><th>Bits</th><th>Default</th><th>Description</th></tr>
<tr><td>Diagnostics (DIAG)</td><td>31:16</td><td>0000h</td><td>Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes:<br>Write ‘1’ clears these bits
<table>
<tr><td>31:27</td><td>Reserved</td></tr>
<tr><td>26</td><td>Exchanged (X): When set to one this bit indicates a COMINIT signal was received. This bit is reflected in the P0IS.PCS bit.</td></tr>
<tr><td>25</td><td>Unknown FIS Type (F): Indicates that one or more FISs were received by the Transport layer with good CRC, but had a type field that was not recognized/known.</td></tr>
<tr><td>24</td><td>Transport state transition error (T): Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared. This bit is always 0 in current implementation.</td></tr>
<tr><td>23</td><td>Link Sequence Error (S): Indicates that one or more Link state machine error conditions was encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition. This bit is always 0 in current implementation.</td></tr>
<tr><td>22</td><td>Handshake Error (H): Indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.</td></tr>
<tr><td>21</td><td>CRC Error (C): Indicates that one or more CRC errors occurred with the Link Layer.</td></tr>
<tr><td>20</td><td>Disparity Error (D): This field is not used by AHCI. This bit is always 0 in current implementation.</td></tr>
<tr><td>19</td><td>10B to 8B Decode Error (B): Indicates that one or more 10B to 8B decoding errors occurred.</td></tr>
<tr><td>18</td><td>Comm Wake (W): Indicates that a Comm Wake signal was detected by the Phy.</td></tr>
<tr><td>17</td><td>Phy Internal Error (I): Indicates that the Phy detected some internal error. This bit is always 0 in current implementation.</td></tr>
<tr><td>16</td><td>PhyRdy Change (N): Indicates that the PhyRdy signal changed state. This bit is reflected in the P0IS.PRCS bit.</td></tr>
</table></td></tr>
</table>

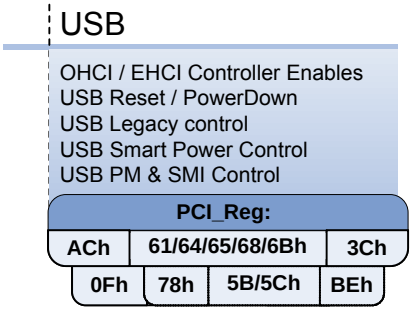
| Port-N Serial ATA Active – RW – 32 bits [Mem_reg: ABAR + port offset + 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device Status (DS) | 31:0 | 00000000h | This field is bit significant. Each bit corresponds to the TAG and command slot of a native queued command, where bit 0 corresponds to TAG 0 and command slot 0. This field is set by software prior to issuing a native queued command for a particular command slot. Prior to writing PxCI[TAG] to '1', software will set DS[TAG] to '1' to indicate that a command with that TAG is outstanding. The device clears bits in this field by sending a Set Device Bits FIS to the host. The HBA clears bits in this field that are set to '1' in the SActive field of the Set Device Bits FIS. The HBA only clears bits that correspond to native queued commands that have completed successfully.<br>Software should only write this field when PxCMD.ST is set to '1'. This field is cleared when PxCMD.ST is written from a '1' to a '0' by software. This field is not cleared by a COMRESET or a software reset. |

| Port-N Command Issue – RW – 32 bits [Mem_reg: ABAR + port offset + 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Commands Issued (CI) | 31:0 | 00000000h | This field is bit significant. Each bit corresponds to a command slot, where bit 0 corresponds to command slot 0. This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field shall only be set to '1' by software when PxCMD.ST is set to '1'.<br>This field is also cleared when PxCMD.ST is written from a '1' to a '0' by software. |

| Port- N SNotification – RWC – 32 bits [Mem_reg: ABAR + port offset + 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| **PM Notify (**PMN**)** | 15:0 | 0000h | This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br><br>PM Port 0h sets bit 0<br>…<br>PM Port Fh sets bit 15<br><br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br><br>This field is reset to default on a HBA Reset, but it is not reset by COMRESET or software reset. |
| Reserved | 31:16 | | Reserved |

## 2.2 OCHI USB 1.1 and EHCI USB 2.0 Controllers

Note: Some USB functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these USB functions and the associated registers.




### *2.2.1 OHCI Registers (Device 19, Function 0, 1, 2, 3, 4)*

#### *2.2.1.1 PCI Configuration Registers (PCI_Reg)*

There are 5 Open-HCI compatible USB host controllers present (functions 0, 1, 2, 3, and 4), and each has their own set of registers. This section describes the configuration registers necessary for the OpenHCI-compliant USB Host Controllers to interface with other system components in a PCI-based PC host. These registers are accessed for set-up during PCI initialization or through special cycles during normal system runtime. Below is summary of the registers that are necessary for the OpenHCI-compliant USB Host Controller to be successfully configured in a PCI-based PC host.

**OHCI0 – PCI config**

| Register Name | Offset Address |
|---|---|
| Device / Vendor ID | 00h |
| Command | 04h |
| Status | 06h |
| Revision ID / Class Code | 08h |
| Miscellaneous | 0Ch |
| BAR_OHCI | 10h |
| Subsystem Vendor ID / Subsystem ID | 2Ch |
| Capability Pointer | 34h |
| Interrupt Line | 3Ch |
| Config Timers / MSI Disable | 40h – 41h |
| Port Disable Control | 42h – 43h |
| OHCI Misc Control | 50h |
| Over Current Control 1 | 58h |
| Over Current Control 2 | 5Ch |
| OHCI OverCurrent Enable | 68h – 69h |
| Target Timeout Control | 74h |
| MSI Control | D0h |
| MSI Address | D4h |
| MSI Data | D8h |
| HT MSI Support | E4h |

**OHCI1/2/3/4 – PCI config**

| Register Name | Offset Address |
|---|---|
| Device / Vendor ID | 00h |
| Command | 04h |
| Status | 06h |
| Revision ID / Class Code | 08h |
| Miscellaneous | 0Ch |
| BAR_OHCI | 10h |
| Subsystem Vendor ID / Subsystem ID | 2Ch |
| Capability Pointer | 34h |
| Interrupt Line | 3Ch |
| MSI Control | D0h |
| MSI Address | D4h |
| MSI Data | D8h |

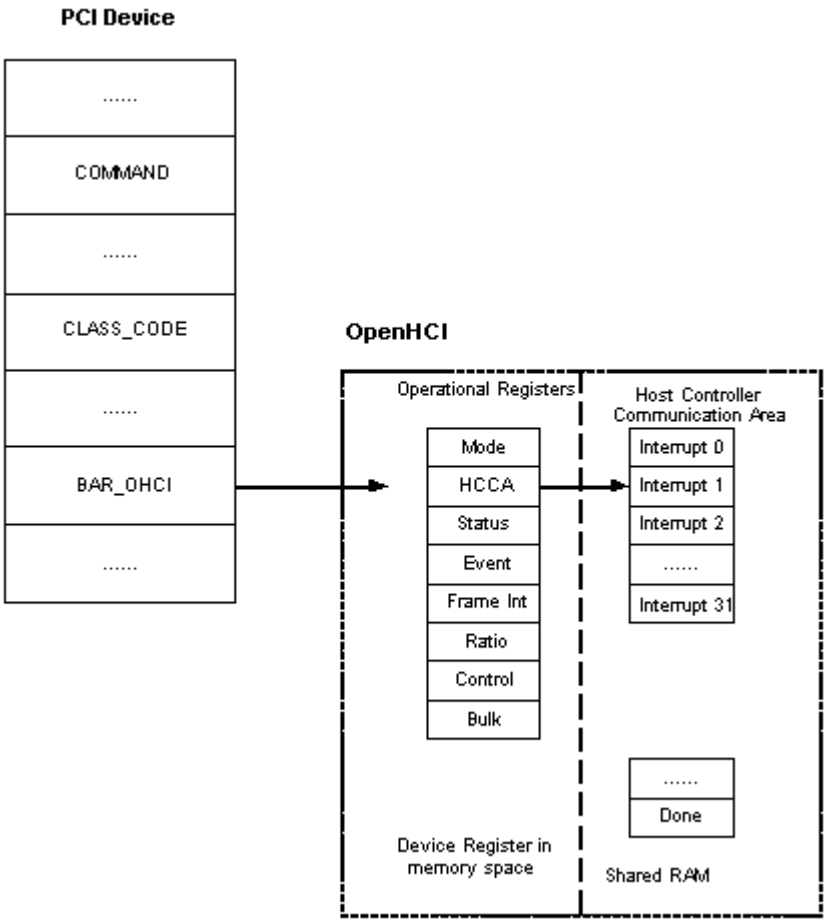



**Figure 3 PCI Configuration Spaces for OHCI**

There are 5 OHCI compatible USB host controllers present (functions 0, 1, 2, 3 and 4), and each has their own set of registers.

| Device / Vendor ID – R - 32 bits - [PCI_Reg : 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VEND_ID | 15:0 | 1002h | Vendor ID |

| Device / Vendor ID – R - 32 bits - [PCI_Reg : 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DEV_ID | 31:16 | Function 0: 4387h<br>Function 1: 4388h<br>Function 2: 4389h<br>Function 3: 438Ah<br>Function 4: 438Bh | Device ID |

| Command – RW - 16 bits - [PCI_Reg : 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Space Accesses | 0 | 0b | A value of 0 disables the device response.<br>A value of 1 allows the device to respond to I/O Space accesses. |
| Memory Space Accesses | 1 | 0b | A value of 0 disables the device response.<br>A value of 1 allows the device to respond to Memory Space accesses. |
| Bus Master | 2 | 0b | A value of 0 disables the device from generating PCI accesses.<br>A value of 1 allows the device to behave as a bus master. |
| Special Cycle | 3 | 0b | Hard-wired to 0, indicating no Special Cycle support. |
| Memory Write and Invalidate Command | 4 | 0b | When it is 0, Memory Write must be used.<br>When it is 1, masters may generate the command. |
| VGA Palette Register Accesses | 5 | 0b | Hard-wired to 0, indicating the device should treat palette write accesses like all other accesses. |
| Parity Enable | 6 | 0b | When it is 1, the device must take its normal action when a parity error is detected.<br>When it is 0, the device sets its Detected Parity Error status bit (bit 15 in the Status register) when an error is detected, but continues normal operations without asserting PERR#. |
| Reserved | 7 | 0b | Hard-wired to 0 per PCI2.3 spec. |
| SERR# Enable | 8 | 0b | A value of 0 disables the SERR# driver.<br>A value of 1 enables the SERR# driver.<br>Address parity errors are reported only if this bit and bit [6] are 1. |
| Fast Back-to-Back Enable | 9 | 0b | A value of 0 means that only fast back-to-back transactions to the same agent are allowed.<br>A value of 1 means the master is allowed to generate fast back-to-back transactions to different agents. |
| Interrupt Disable | 10 | 0b | A value of 0 enables the assertion of the device/function's INTx# signal.<br>A value of 1 disables the assertion of the device/function's INTx# signal. |
| Reserved | 15:11 | | Reserved |

| Status – R - 16 bits - [PCI_Reg : 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | | Reserved |
| Interrupt Status | 3 | 0b | This bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1 will the device's/function's INTx# signal be asserted. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit. |
| Capabilities List | 4 | 1b | A value of 0 indicates that no New Capabilities linked list is available.<br>A value of 1 indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 66 MHz Capable | 5 | 1b | Hard-wired to 1, indicating 66MHz capable. |
| Reserved | 6 | | Reserved |
| Fast Back-to-Back Capable | 7 | 1b | Hard-wired to 1, indicating Fast Back-to-Back capable. |

| Status – R - 16 bits - [PCI_Reg : 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Master Data Parity Error | 8 | 0b | This bit is set only when three conditions are met: 1) the bus agent asserted PERR# itself (on a read) or observed PERR# asserted (on a write); 2) the agent setting the bit acted as the bus master for the operation in which the error occurred; and 3) the Parity Error Response bit (Command register) is set. |
| DEVSEL timing | 10:9 | 01b | Hard-wired to 01b – medium timing |
| Signaled Target Abort | 11 | 0b | This bit is set by a target device whenever it terminates a transaction with Target-Abort. |
| Received Target Abort | 12 | 0b | This bit is set by a master device whenever its transaction is terminated with Target-Abort. |
| Received Master Abort | 13 | 0b | This bit is set by a master device whenever its transaction (except for Special Cycle) is terminated with Master-Abort. |
| Signaled System Error | 14 | 0b | This bit is set whenever the device asserts SERR#. |
| Detected Parity Error | 15 | 0b | This bit is set by the device whenever it detects a parity error, even if parity error handling is disabled (as controlled by bit 6 in the Command register). |

| Revision ID / Class Code – R - 32 bits - [PCI_Reg : 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | Revision ID. |
| PI | 15:8 | 10h | Programming Interface. A constant value of '10h' indentifies the device being an OpenHCI Host Controller. |
| SC | 23:16 | 03h | Sub Class. A constant value of '03h' indentifies the device being of Universal Serial Bus. |
| BC | 31:24 | 0Ch | Base Class. A constant value of '0Ch' identifies the device being a Serial Bus Controller. |

| Miscellaneous – RW/R - 32 bits - [PCI_Reg : 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | This read/write field specifies the system cacheline size in units of DWORDs and must be initialized to 00h. |
| Latency Timer | 15:8 | 00h | [9:8] hard-wired to 00b, resulting in a timer granularity of at least four clocks. This field specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. |
| Header Type | 23:16 | 80h/00h | This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space) and also whether or not the device contains multiple functions.<br><br>Function 0: Bit[23] hard-wired to 1 → the device has multiple functions.<br>Function 1: Bit[23] hard-wired to 0 → the device is single function.<br>Function 2: Bit[23] hard-wired to 0 → the device is single function.<br>Function 3: Bit[23] hard-wired to 0 → the device is single function.<br>Function 4: Bit[23] hard-wired to 0 → the device is single function.<br><br>Bits [22:16] are hard-wired to 00h. |
| BIST | 31:24 | 00h | Hard-wired to 00h, indicating no build-in BIST support. |

| Bar_OHCI – RW - 32 bits - [PCI_Reg : 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IND | 0 | 0b | Indicator. A constant value of '0' indicates that the operational registers of the device are mapped into memory space of the main memory of the PC host system. Read Only. |
| TP | 2:1 | 0h | Type. A constant value of '00b' indicates that the base register is 32-bit wide and can be placed anywhere in the 32-bit memory space; i.e., lower 4 GB of the main memory of the PC host. Read Only. |

| Bar_OHCI – RW - 32 bits - [PCI_Reg : 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PM | 3 | 0b | Prefetch memory. A constant value of '0' indicates that there is no support for "prefetchable memory". Read Only. |
| | 11:4 | 00h | Represents a maximum of 4-KB addressing space for the OpenHCi's operational registers. Read Only. |
| BAR | 31:12 | 000h | Base Address. Specifies the upper 20 bits of the 32-bit starting base address. This represent a maximum of 4-KB addressing space for the OpenHCI's operational registers. |

| Subsystem Vendor ID / Subsystem ID – RW - 32 bits - [PCI_Reg : 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Can only be written once by software. |
| Subsystem ID | 31:16 | 0000h | Can only be written once by software. |

| Capability Pointer – R - 8 bits - [PCI_Reg : 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability Pointer | 7:0 | D0h | Address of the 1st element of capability link. |

| Interrupt Line – RW – 32 bits - [PCI_Reg : 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | The Interrupt Line register is an eight-bit register used to communicate interrupt line routing information. The register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST software will write the routing information into this register as it initializes and configures the system.<br><br>The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. The device itself does not use this value; rather it is used by device drivers and operating systems. Device drivers and operating systems can use this information to determine priority and vector information. Values in this register are system architecture specific. |
| Interrupt Pin | 15:8 | <br>01h<br>02h<br>03h<br>02h<br>03h | Read Only by default.<br>OHCI0: Hard-wired to 01h, corresponding to using INTA#.<br>OHCI1: Hard-wired to 02h, corresponding to using INTB#.<br>OHCI2: Hard-wired to 03h, corresponding to using INTC#.<br>OHCI3: Hard-wired to 02h, corresponding to using INTB#.<br>OHCI4: Hard-wired to 03h, corresponding to using INTC#. |
| MIN_GNT | 23:16 | 00h | Read Only. Hardwired to 00h to indicate no major requirements for the settings of Latency Timers. |
| MAX_LAT | 31:24 | 00h | Read Only. Hardwired to 00h to indicate no major requirements for the settings of the Latency Timers. |

| Config Timers / MSI Disable (OHCI0 only) – RW - 16 bits - [PCI_Reg : 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TRDY Timer | 7:0 | 80h | Target Ready timer to timeout non-responding target. |

| Config Timers / MSI Disable (OHCI0 only) – RW - 16 bits - [PCI_Reg : 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Disable | 12:8 | 00h | When these bits are set MSI capability will be disabled for the corresponding Host Controller.<br>Bit 8 – OHCI0<br>Bit 9 – OHCI1<br>Bit 10 – OHCI2<br>Bit 11 – OHCI3<br>Bit 12 – OHCI4 |
| Reserved | 15:13 | 0h | Reserved |

| Port Disable (OHCI0 only) – RW - 16 bits - [PCI_Reg : 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port_disable | 9:0 | 00h | When these bits are set the corresponding ports are disabled. For example, if bit-0 is set, then port-0 (its corresponding port) is disabled; if bit-1 is set, then its corresponding port (port-1) is disabled (and so on). |
| Reserved | 15:10 | 00h | Reserved |

| OHCI Misc Control (OHCI0 only) – RW - 16 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | Reserved. |
| DisUsbS3OvrCur | 8 | 0b | Set to 1 to disable over-current detection for both EHCII and OHCI. |
| Reserved | 9 | 0b | Reserved. |
| OHCI Cache Enable | 10 | 1b | Enable bit for 64 byte OHCI DMA cache. |
| OHCI Prefetch Enable | 11 | 1b | Enable bit to prefetch next cache line for OHCI DMA reads. |
| SMI Handshake Disable | 12 | 0b | If this bit is set the Handshake between USB and ACPI is disabled when SMI is requested by USB |
| Reserved | 15: 13 | 0h | Reserved. |

| Over Current Control 1 (OHCI0 only) – R - 32 bits - [PCI_Reg : 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port0 OverCurrent Control | 3:0 | Fh | The register is to control the OverCurrent pin mapping for Port-0. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-0. |
| Port1 OverCurrent Control | 7:4 | Fh | The register is to control the OverCurrent pin mapping for Port-1. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-1. |
| Port2 OverCurrent Control | 11:8 | Fh | The register is to control the OverCurrent pin mapping for Port-2. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-2. |
| Port3 OverCurrent Control | 15:12 | Fh | The register is to control the OverCurrent pin mapping for Port-3. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-3. |
| Port4 OverCurrent Control | 19:16 | Fh | The register is to control the OverCurrent pin mapping for Port-4. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-4. |

| Over Current Control 1 (OHCI0 only) – R - 32 bits - [PCI_Reg : 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port5 OverCurrent Control | 23:20 | Fh | The register is to control the OverCurrent pin mapping for Port-5. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-5. |
| Port6 OverCurrent Control | 27:24 | Fh | The register is to control the OverCurrent pin mapping for Port-6. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-6. |
| Port7 OverCurrent Control | 31:28 | Fh | The register is to control the OverCurrent pin mapping for Port-7. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-7. |
| There are 10 pins can be used as USB OverCurrent function –<br>USB_OC0#/GPM0# USB_OC1#/GPM1# USB_OC2#/GPM2# USB_OC3#/GPM3#<br>USB_OC4#/GPM4# USB_OC5#/GPM5# USB_OC6#/GEVENT6# USB_OC7#/GEVENT7#<br>USB_OC8#/GPM8# USB_OC9#/SLP_S2/GPM9#<br><br>Register value-to-OverCurrent Pin mapping:<br>USB_OC0# = 0000, USB_OC1# = 0001, USB_OC2# = 0010, USB_OC3# = 0011,<br>USB_OC4# = 0100, USB_OC5# = 0101, USB_OC6# = 0110, USB_OC7# = 0111,<br>USB_OC8# = 1000, USB_OC9# = 1001<br><br>* Note: Since OverCurrent pins can be used as GPM# as well, the corresponding register bits to set the pin as OverCurrent have to be set in Smbus Controller. | | | |

| Over Current Control 2 (OHCI0 only) – R - 32 bits - [PCI_Reg : 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port8 OverCurrent Control | 3:0 | Fh | The register is to control the OverCurrent pin mapping for Port-8. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-8. |
| Port9 OverCurrent Control | 7:4 | Fh | The register is to control the OverCurrent pin mapping for Port-9. There are 10 OverCurrent pins (USB_OC0 ~ USB_OC9), any value greater than 0x9h will disable the OverCurrent function for port-9. |
| Reserved | 31:8 | | Reserved |
| There are 10 pins can be used as USB OverCurrent function –<br>USB_OC0#/GPM0# USB_OC1#/GPM1# USB_OC2#/GPM2# USB_OC3#/GPM3#<br>USB_OC4#/GPM4# USB_OC5#/GPM5# USB_OC6#/GEVENT6# USB_OC7#/GEVENT7#<br>USB_OC8#/GPM8# USB_OC9#/SLP_S2/GPM9#<br><br>Register value-to-OverCurrent Pin mapping:<br>USB_OC0# = 0000, USB_OC1# = 0001, USB_OC2# = 0010, USB_OC3# = 0011,<br>USB_OC4# = 0100, USB_OC5# = 0101, USB_OC6# = 0110, USB_OC7# = 0111,<br>USB_OC8# = 1000, USB_OC9# = 1001<br><br>*Note: Since OverCurrent pins can be used as GPM# as well, the corresponding register bits to set the pin as OverCurrent have to be set in Smbus Controller. | | | |

| OHCI OverCurrent Enable (OHCI0 only) – RW - 16 bits - [PCI_Reg : 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| OHCI OverCurrent Enable | 9:0 | 00h | Writing this bit to a one enables the respective port to be sensitive to over-current conditions as wake-up events when it is owned by OHCI. |
| Reserved | 15:10 | 00h | Reserved |

| Target Timeout Control (OHCI0 only) – RW - 32 bits - [PCI_Reg : 74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Retry counter | 7:0 | FFh | Counter to control the purge of the delay queue when the host controller does not return the ack. After the counter expires the transaction is target aborted.<br>The retry counter can be disabled by writing 00h in this Register. |
| Reserved | 23:8 | 0000h | Reserved |
| Timeout Timer | 31:24 | 80h | Timer to control the purge of the delay queue when the master that has initiated the access does not return to complete the transaction. After the timer expires the queue is invalidated and the next transaction is serviced. |

| MSI Control – RW - 32 bits - [PCI_Reg : D0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI USB | 7:0 | 05h | MSI USB ID. Read only. |
| Next Item Pointer | 15:8 | 00h | Pointer to next capability structure |
| MSI Control Out | 16 | 0b | Set to 1 to disable IRQ. Use MSI instead. |
| Reserved | 19:17 | 0h | Reserved |
| MSI Control | 22:20 | 0h | MSI control field |
| Reserved | 31:23 | 00h | Reserved |

| MSI Address – RW - 32 bits - [PCI_Reg : D4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Address | 31:0 | 0h | System-specified message address. |

| MSI Data – RW - 16 bits - [PCI_Reg : D8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Data | 15:0 | 0h | System-specified message. |

#### *2.2.1.2 OHCI Operational Registers (MEM_Reg)*

The Host Controller (HC) contains a set of on-chip operational registers, which are mapped into a non-cacheable portion of the system addressable space. These registers are used by the Host Controller Driver (HCD) and should be read and written as Dwords.

Reserved bits may be allocated in future releases and should not be assumed to contain 0. The Host Controller Driver should always preserve the value(s) of the reserved field. When a write to set/clear register is written, bits written to reserved fields should be 0.

| Register Name | Offset Address |
|---|---|
| HcRevision | 0h |
| HcControl | 4h |
| HcCommandStatus | 8h |
| HcInterruptStatus | Ch |
| HcInterruptEnable | 10h |
| HcInterruptDisable | 14h |
| HcHCCA | 18h |
| HcPeriodCurrentED | 1Ch |
| HcControlHeadED | 20h |
| HcControlCurrentED | 24h |
| HcBulkHeadED | 28h |

| Register Name | Offset Address |
|---|---|
| HcBulkCurrentED | 2Ch |
| HcDoneHead | 30h |
| HcFmInterval | 34h |
| HcFmRemaining | 38h |
| HcFmNumber | 3Ch |
| HcPeriodicStart | 40h |
| HcLSThreshold | 44h |
| HcRhDescriptorA | 48h |
| HcRhDescriptorB | 4Ch |
| HCRhStatus | 50h |
| HcRhPortStatus[1] | 54h |
| … | … |
| HcRhPortStatus[NDP] | 54+4*NDP |

| HcRevision – R - 32 bits - [MEM_Reg : 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| REV | 7:0 | 10h | **Revision.**<br>This read-only field contains the version of HCI specification. |
| L | 8 | 1b | **Legacy**<br>This read-only field is 1, indicating that the legacy support registers are present in this HC. |
| Reserved | 31:9 | | Reserved |

| HcControl - 32 bits - [MEM_Reg : 04h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| CBSR | 1:0 | 00b | RW | R | **ControlBulkServiceRatio**<br>This specifies the service ratio between Control and Bulk Eds. Before processing any of the non-periodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control Eds have been processed, in determining whether to continue serving another Control ED or switching to Bulk Eds.<br>CBSR<br>No. of Control Eds Over Bulk Eds Served<br><br>0<br>1:1<br><br>1<br>2:1<br><br>2<br>3:1<br><br>3<br>4:1 |
| PLE | 2 | 0b | RW | R | **PeriodicListEnable**<br>This bit is set to enable the processing of the periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list. |

| HcControl - 32 bits - [MEM_Reg : 04h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| IE | 3 | 0b | RW | R | **IsochronousEnable**<br>This bit is used by HCD to enable/disable processing of isochronous Eds. While processing the periodic list in a Frame, HC checks he status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the Eds. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous Eds) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame). |
| CLE | 4 | 0b | RW | R | **ControlListEnable**<br>This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcControlCurrentED is pointing to an ED to be removed, HCD must advance the pointer by updating HcControlCurrentED before re-enabling processing of the list. |
| BLE | 5 | 0b | RW | R | **BulkListEnable**<br>This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If *HcBulkCurrentED* is pointing to an ED to be removed, HCD must advance the pointer by updating *HcBulkCurrentED* before re-enabling processing of the list. |
| HCFS | 7:6 | 00b | RW | RW | **HostControllerFunctionalState** for USB<br>00b: USBRESET<br>01b: USBRESUME<br>10b: USBOPERATIONAL<br>11b: USBSUSPEND<br>A transition to USBOPERATIONAL from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the **StartofFrame** field of *HcInterruptStatus*.<br>This field may be changed by HC only when in the USBSUSPEND state. HC may move from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port.<br>HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports. |

| HcControl - 32 bits - [MEM_Reg : 04h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| IR | 8 | 0b | RW | R | **InterruptRouting**<br>This bit determines the routing of interrupts generated by events registered in *HcInterruptStatu*s. If clear, all interrupts are routed to the normal host bus interrupt mechanism. If set, interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC. |
| RWC | 9 | 0b | RW | RW | **RemoteWakeupConnected**<br>This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification. |
| RWE | 10 | 0b | RW | R | **RemoteWakeupEnable**<br>This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the *ResumeDetected* bit in *HcInterruptStatus* is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt. |
| Reserved | 31:11 | | | | Reserved |

| HcCommandStatus - 32 bits - [MEM_Reg : 08h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| HCR | 0 | 0b | RW | RW | **HostControllerReset**<br>This bit is set by HCD to initiate a software reset of HC.Regardless of the functional state of HC, it moves to the USBSUSPEND state in which most of the operational registers are reset except those stated otherwise; e.g., the *InterruptRouting* field of *HcContro*l, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 ms. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports. |

| HcCommandStatus - 32 bits - [MEM_Reg : 08h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| CLF | 1 | 0b | RW | RW | **ControlListFilled**<br>This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list. When HC begins to process the head of the Control list, it checks CLF. As long as *ControlListFilled* is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set *ControlListFilled* to 0. If HC finds a TD on the list, then HC will set *ControlListFilled* to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set *ControlListFilled*, then *ControlListFilled* will still be 0 when HC completes processing the Control list and Control list processing will stop. |
| BLF | 2 | 0b | RW | RW | **BulkListFilled**<br>This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BF. As long as *BulkListFilled* is 0, HC will not start processing the Bulk list. If *BulkListFilled* is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set *BulkListFilled* to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set *BulkListFilled*, then *BulkListFilled* will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop. *BulkListFilled*<br>This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BF. As long as *BulkListFilled* is 0, HC will not start processing the Bulk list. If *BulkListFilled* is 1, HC will start processing the<br>Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set *BulkListFilled* to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set *BulkListFilled*, then *BulkListFilled* will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop. |
| OCR | 3 | 0b | RW | RW | **OwnershipChangeRequest**<br>This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the **OwnershipChange** field in *HcInterruptStatu*s. After the changeover, this bit is cleared and remains so until the next request from OS HCD. |
| Reserved | 15:4 | | | | |
| SOC | 17:16 | 00b | R | RW | **SchedulingOverrunCount**<br>These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if **SchedulingOverrun** in *HcInterruptStatus* has already been set. This is used by HCD to monitor any persistent scheduling problems. |
| Reserved | 31:18 | | | | Reserved |

| HcInterruptStatus – RW - 32 bits - [MEM_Reg : 0Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| SO | 0 | 0b | RW | RW | **SchedulingOverrun**<br>This bit is set when the USB schedule for the current Frame overruns and after the update of *HccaFrameNumbe*r. A scheduling overrun will also cause the **SchedulingOverrunCount** of *HcCommandStatus* to be incremented. |
| WDH | 1 | 0b | RW | RW | **WritebackDoneHead**<br>This bit is set immediately after HC has written *HcDoneHead* to *HccaDoneHea*d. Further updates of the *HccaDoneHead* will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of *HccaDoneHea*d. |
| SF | 2 | 0b | RW | RW | **StartofFrame**<br>This bit is set by HC at each start of a frame and after the update of *HccaFrameNumbe*r. HC also generates a SOF token at the same time. |
| RD | 3 | 0b | RW | RW | **ResumeDetected**<br>This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRESUME state. |
| UE | 4 | 0b | RW | RW | **UnrecoverableError**<br>This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset. |
| FNO | 5 | 0b | RW | RW | **FrameNumberOverflow**<br>This bit is set when the MSb of *HcFmNumber* (bit 15) changes value, from 0 to 1 or from 1 to 0, and after *HccaFrameNumber* has been updated. |
| RHSC | 6 | 0b | RW | RW | **RootHubStatusChange**<br>This bit is set when the content of *HcRhStatus* or the content of any of *HcRhPortStatu*s [**NumberofDownstreamPor**t] has changed. |
| Reserved | 29:7 | | | | Reserved |
| OC | 30 | 0b | RW | RW | **OwnershipChange**<br>This bit is set by HC when HCD sets the **OwnershipChangeRequest** field in *HcCommandStatu*s. This event, when unmasked, will always generate an System Management Interrupt (SMI) immediately.<br>This bit is tied to 0b when the SMI pin is not implemented. |
| Reserved | 31 | | | | Reserved |

| HcInterruptEnable - 32 bits - [MEM_Reg : 10h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| SO | 0 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Scheduling Overrun. |
| WDH | 1 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to HcDoneHead Writeback. |
| SF | 2 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Start of Frame. |

| HcInterruptEnable - 32 bits - [MEM_Reg : 10h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| RD | 3 | 0b | RW | W | 0 - Ignore<br>1 - Enable interrupt generation due to Resume Detect. |
| UE | 4 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Unrecoverable Error. |
| FNO | 5 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Frame Number Overflow. |
| RHSC | 6 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Root Hub Status Change. |
| Reserved | 29:7 | | | | Reserved |
| OC | 30 | 0b | RW | RW | 0 - Ignore<br>1 - Enable interrupt generation due to Ownership Change. |
| MIE | 31 | 0b | RW | R | A ‘0’ written to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as a Master Interrupt Enable. |

| HcInterruptDisable - 32 bits - [MEM_Reg : 14h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| SO | 0 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Scheduling Overrun. |
| WDH | 1 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to HcDoneHead Writeback. |
| SF | 2 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Start of Frame. |
| RD | 3 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Resume Detect. |
| UE | 4 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Unrecoverable Error. |
| FNO | 5 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Frame Number Overflow. |
| RHSC | 6 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Root Hub Status Change. |
| Reserved | 29:7 | | | | Reserved |
| OC | 30 | 0b | RW | R | 0 - Ignore<br>1 - Disable interrupt generation due to Ownership Change. |
| MIE | 31 | 0b | RW | R | A '0' written to this field is ignored by HC. A '1' written to this field disables interrupt generation due to events specified in the other bits of this register. This field is set after a hardware or software reset. |

| HcHCCA - 32 bits - [MEM_Reg : 18h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 7:0 | | | | Reserved |
| HCCA | 31:8 | 000000h | RW | R | This is the base address of the Host Controller Communication Area |

| HcPeriodCurrentED - 32 bits - [MEM_Reg : 1Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |
| PCED | 31:4 | 0000000h | R | RW | **PeriodCurrentED**<br>This is used by HC to point to the head of one of the Periodic lists which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading. |

| HcControlHeadED- 32 bits - [MEM_Reg : 20h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |
| CHED | 31:4 | 0000000h | RW | R | **ControlHeadED**<br>HC traverses the Control list starting with the *HcControlHeadED* pointer. The content is loaded from HCCA during the initialization of HC. |

| HcControlCurrentED - 32 bits - [MEM_Reg : 24h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |
| CCED | 31:4 | 0000000h | RW | RW | **ControlCurrentED**<br>This pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the **ControlListFilled** of in *HcCommandStatu*s. If set, it copies the content of *HcControlHeadED* to *HcControlCurrentED* and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the **ControlListEnable** of *HcControl* is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list. |

| HcBulkHeadED - 32 bits - [MEM_Reg : 28h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |
| BHED | 31:4 | 0b | RW | R | **BulkHeadED**<br>HC traverses the Bulk list starting with the *HcBulkHeadED* pointer. The content is loaded from HCCA during the initialization of HC. |

| HcBulkCurrentED - 32 bits - [MEM_Reg : 2Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |

| HcBulkCurrentED - 32 bits - [MEM_Reg : 2Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| BCED | 31:4 | 0000000h | RW | RW | **BulkCurrentED**<br>This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the **ControlListFilled** of HcControl. If set, it copies the content of *HcBulkHeadED* to *HcBulkCurrentED* and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the **BulkListEnable** of *HcControl* is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list. |

| HcDoneHead - 32 bits - [MEM_Reg : 30h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| Reserved | 3:0 | | | | Reserved |
| DH | 31:4 | 0b | R | RW | **DoneHead**<br>When a TD is completed, HC writes the content of *HcDoneHead* to the NextTD field of the TD. HC then overwrites the content of *HcDoneHead* with the address of this TD. This is set to zero whenever HC writes the content of this register to HCCA. It also sets the **WritebackDoneHead** of *HcInterruptStatu*s. |

| HcFmInterval - 32 bits - [MEM_Reg : 34h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| FI | 13:0 | 2EDFh | RW | R | **FrameInterval**<br>This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999.<br>HCD should store the current value of this field before resetting HC. By setting the **HostControllerReset** field of *HcCommandStatus* as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence. |
| Reserved | 15:14 | | | | Reserved |
| FSMPS | 30:16 | 0000h | RW | R | **FSLargestDataPacket**<br>This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD. |
| FIT | 31 | 0b | RW | R | **FrameIntervalToggle**<br>HCD toggles this bit whenever it loads a new value to **FrameInterval** |

| HcFmRemaining - 32 bits - [MEM_Reg : 38h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| FR | 13:0 | 0000h | R | RW | **FrameRemaining**<br>This counter is decremented at each bit time. When it reaches zero, it is reset by loading the **FrameInterval** value specified in *HcFmInterval* at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the **FrameInterval** of *HcFmInterval* and uses the updated value from the next SOF. |
| Reserved | 30:14 | | | | Reserved |
| FRT | 31 | 0b | R | RW | **FrameRemainingToggle**<br>This bit is loaded from the **FrameIntervalToggle** field of *HcFmInterval* whenever **FrameRemaining** reaches 0. This bit is used by HCD for the synchronization between **FrameInterval** and **FrameRemainin**g. |

| HcFmNumber - 32 bits - [MEM_Reg : 3Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| FN | 15:0 | 0000h | R | RW | This is incremented when *HcFmRemaining* is re-loaded. It will be rolled over to 0h after ffffh. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the **FrameNumber** at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the **StartofFrame** in *HcInterruptStatus.* |
| Reserved | 31:16 | | | | Reserved |

| HcPeriodicStart - 32 bits - [MEM_Reg : 40h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| PS | 13:0 | 0000h | RW | R | **PeriodicStart**<br>After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from *HcFmInterva*l. A typical value will be 3E67h. When *HcFmRemaining* reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress. |
| Reserved | 31:14 | | | | Reserved |

| HcLSThreshold - 32 bits - [MEM_Reg : 44h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| LST | 11:0 | 0628h | RW | R | **LSThreshold**<br>This field contains a value which is compared to the **FrameRemaining** field prior to initiating a Low Speed transaction. The transaction is started only if **FrameRemaining**, this field. The value is calculated by HCD with the consideration of transmission and setup overhead. |
| Reserved | 31:12 | | | | Reserved |

| HcRhDescriptorA - 32 bits - [MEM_Reg : 48h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| NDP | 7:0 | 02h | R | R | **NumberDownstreamPorts**<br>These bits specify the number of downstream ports supported by the Root Hub. |
| PSM | 8 | 0b | RW | R | **PowerSwitchingMode**<br>This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the **NoPowerSwitching** field is cleared.<br>0: all ports are powered at the same time.<br>1: each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the **PortPowerControlMask** bit is set, the port responds only to port power commands **(Set/ClearPortPowe**r). If the port mask is cleared, then the port is controlled only by the global power switch **(Set/ClearGlobalPowe**r). |
| NPS | 9 | 1b | RW | R | **NoPowerSwitching**<br>These bits are used to specify whether power switching is supported or port are always powered. It is implementation- specific. When this bit is cleared, the **PowerSwitchingMode** specifies global or per-port switching.<br>0: Ports are power switched<br>1: Ports are always powered on when the HC is powered on |
| DT | 10 | 0b | R | R | **DeviceType**<br>This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0. |
| OCPM | 11 | 1b | RW | R | **OverCurrentProtectionMode**<br>This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, this field should reflect the same mode as **PowerSwitchingMod**e. This field is valid only if the **NoOverCurrentProtection** field is cleared.<br>0: over-current status is reported collectively for all downstream ports<br>1: over-current status is reported on a per-port basis |
| NOCP | 12 | 0b | RW | R | **NoOverCurrentProtection**<br>This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the **OverCurrentProtectionMode** field specifies global or per-port reporting.<br>0: Over-current status is reported collectively for all downstream ports<br>1: No overcurrent protection supported |
| Reserved | 23:13 | | | | Reserved |
| POTPGT | 31:24 | 02h | RW | R | **PowerOnToPowerGoodTime**<br>This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as **POTPGT** * 2 ms. |

| HcRhDescriptorB - 32 bits - [MEM_Reg : 4Ch] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| DR | 15:0 | 0000h | RW | R | **DeviceRemovable**<br>Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable.<br>bit 0: Reserved<br>bit 1: Device attached to Port #1<br>bit 2: Device attached to Port #2<br>...<br>bit15: Device attached to Port #15 |
| PPCM | 31:16 | 0000h | RW | R | **PortPowerControlMask**<br>Each bit indicates if a port is affected by a global power control command when **PowerSwitchingMode** is set. When set, the port's power state is only affected by per-port power control **(Set/ClearPortPowe**r). When cleared, the port is controlled by the global power switch **(Set/ClearGlobalPowe**r). If the device is configured to global switching mode **(PowerSwitchingMod**e=0), this field is not valid.<br>bit 0: Reserved<br>bit 1: Ganged-power mask on Port #1<br>bit 2: Ganged-power mask on Port #2<br>...<br>bit15: Ganged-power mask on Port #15 |

| HcRhStatus - 32 bits - [MEM_Reg : 50h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| LPS | 0 | 0b | RW | R | (Read) **LocalPowerStatus**<br>The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.<br><br>(Write) **ClearGlobalPower**<br>In global power mode **(PowerSwitchingMod**e=0), This bit is written to '1' to turn off power to all ports (Clear) **PortPowerStatu**s). In per-port power mode, it clears **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a '0' has no effect. |
| OCI | 1 | 0b | R | RW | **OverCurrentIndicator**<br>This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented this bit is always '0' |
| Reserved | 14:2 | | | | Reserved |
| DRWE | 15 | 0b | RW | R | (Read) **DeviceRemoteWakeupEnable**<br>This bit enables a **ConnectStatusChange** bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the **ResumeDetected** interrupt.<br>0 = **ConnectStatusChange** is not a remote wakeup event.<br>1 = **ConnectStatusChange** is a remote wakeup event.<br><br>(Write) **SetRemoteWakeupEnable**<br>Writing a '1' sets **DeviceRemoveWakeupEnabl**e.<br>Writing a '0' has no effect. |

| HcRhStatus - 32 bits - [MEM_Reg : 50h] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| LPSC | 16 | 0b | RW | R | (Read) **LocalPowerStatusChange**<br>The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.<br><br>(Write) **SetGlobalPower**<br>In global power mode **(PowerSwitchingMod**e=0), This bit is written to '1' to turn on power to all ports (Clear)<br>**PortPowerStatu**s). In per-port power mode, it sets **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a '0' has no effect. |
| OCIC | 17 | 0b | RW | RW | **OverCurrentIndicatorChange**<br>This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a '1'. Writing a '0' has no effect. |
| Reserved | 30:18 | | | | Reserved |
| CRWE | 31 | - | W | R | (Write) **ClearRemoteWakeupEnable**<br>Writing a '1' clears **DeviceRemoveWakeupEnabl**e. Writing a '0' has no effect. |

| HcRhPortStatus - 32 bits - [MEM_Reg : 50h+4*(1:NDP)] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| CCS | 0 | 0b | RW | RW | (Read) **CurrentConnectStatus**<br>This bit reflects the current state of the downstream port.<br>0 = No device connected<br>1 = Device connected<br><br>(Write) **ClearPortEnable**<br>The HCD writes a '1' to this bit to clear the **PortEnableStatus** bit.<br>Writing a '0' has no effect. The **CurrentConnectStatus** is not affected by any write.<br>Note: This bit is always read '1b' when the attached device is non-removable **(DeviceRemoveable[NDP**]). |

| HcRhPortStatus - 32 bits - [MEM_Reg : 50h+4*(1:NDP)] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| PES | 1 | 0b | RW | RW | (Read) **PortEnableStatus**<br>This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes **PortEnabledStatusChange** to be set. HCD sets this bit by writing **SetPortEnable** and clears it by writing **ClearPortEnabl**e. This bit cannot be set when **CurrentConnectStatus** is cleared. This bit is also set, if not already, at the completion of a port reset when **ResetStatusChange** is set or port suspend when **SuspendStatusChange** is set.<br><br>0 = Port is disabled<br>1 = Port is enabled<br><br>(Write) **SetPortEnable**<br>The HCD sets **PortEnableStatus** by writing a '1'. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortEnableStatu**s, but instead sets **ConnectStatusChang**e. This informs the driver that it attempted to enable a disconnected port. |
| PSS | 2 | 0b | RW | RW | (Read) **PortSuspendStatus**<br>This bit indicates the port is suspended or in the resume sequence. It is set by a **SetSuspendState** write and cleared when **PortSuspendStatusChange** is set at the end of the resume interval. This bit cannot be set if **CurrentConnectStatus** is cleared. This bit is also cleared when **PortResetStatusChange** is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC.<br>0 = Port is not suspended<br>1 = Port is suspended<br><br>(Write) **SetPortSuspend**<br>The HCD sets the **PortSuspendStatus** bit by writing a '1' to this bit. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortSuspendStatu**s; instead it sets **ConnectStatusChang**e. This informs the driver that it attempted to suspend a disconnected port. |
| POCI | 3 | 0b | RW | RW | (Read) **PortOverCurrentIndicator**<br>This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal<br>0 = No overcurrent condition.<br>1 = Overcurrent condition detected.<br>(Write) **ClearSuspendStatus**<br>The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if **PortSuspendStatus** is set. |

| HcRhPortStatus - 32 bits - [MEM_Reg : 50h+4*(1:NDP)] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| PRS | 4 | 0b | RW | RW | (Read) **PortResetStatus**<br>When this bit is set by a write to **SetPortRese**t, port reset signaling is asserted. When reset is completed, this bit is cleared when **PortResetStatusChange** is set. This bit cannot be set if **CurrentConnectStatus** is cleared.<br>0 = Port reset signal is not active<br>1 = Port reset signal is active<br>(Write) **SetPortReset**<br>The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortResetStatu**s, but instead sets **ConnectStatusChang**e. This informs the driver that it attempted to reset a disconnected port. |
| Reserved | 7:5 | | | | Reserved |
| PPS | 8 | 0b | RW | RW | (Read) **PortPowerStatus**<br>This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing **SetPortPower** or **SetGlobalPowe**r. HCD clears this bit by writing **ClearPortPower** or **ClearGlobalPowe**r. Which power control switches are enabled is determined by **PowerSwitchingMode** and **PortPortControlMask[NDP**]. In global switching mode, **(PowerSwitchingMod**e=0), only **Set/ClearGlobalPower** controls this bit. In per-port power switching **(PowerSwitchingMod**e=1), if the **PortPowerControlMask[NDP]** bit for the port is set, only **Set/ClearPortPower** commands are enabled. If the mask is not set, only **Set/ClearGlobalPower** commands are enabled. When port power is disabled, **CurrentConnectStatu**s, **PortEnableStatu**s, **PortSuspendStatu**s, and **PortResetStatus** should be reset.<br>0 = Port power is off<br>1 = Port power is on<br>(Write) **SetPortPower**<br>The HCD writes a '1' to set the **PortPowerStatus** bit. Writing a '0' has no effect.<br>Note: This bit is always reads '1b' if power switching is not supported. |
| LSDA | 9 | X | RW | RW | (Read) **LowSpeedDeviceAttached**<br>This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the **CurrentConnectStatus** is set.<br>0 = Full speed device attached<br>1 = Low speed device attached<br>(Write) **ClearPortPower**<br>The HCD clears the **PortPowerStatus** bit by writing a '1' to this bit. Writing a '0' has no effect. |
| Reserved | 15:10 | | | | Reserved |

| HcRhPortStatus - 32 bits - [MEM_Reg : 50h+4*(1:NDP)] | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **HCD** | **HC** | **Description** |
| CSC | 16 | 0b | RW | RW | **ConnectStatusChange**<br>This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared when a **SetPortReset, SetPortEnabl**e, or **SetPortSuspend** write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected.<br>0 = No change in **CurrentConnectStatus**<br>1 = Change in **CurrentConnectStatus**<br>Note: If the **DeviceRemovable[NDP]** bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached. |
| PESC | 17 | 0b | RW | RW | **PortEnableStatusChange**<br>This bit is set when hardware events cause the **PortEnableStatus** bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br>0 = No change in **PortEnableStatus**<br>1 = Change in **PortEnableStatus** |
| PSSC | 18 | 0b | RW | RW | **PortSuspendStatusChange**<br>This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resychronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when **ResetStatusChange** is set.<br>0 = Resume is not completed<br>1 = Resume completed |
| OCIC | 19 | 0b | RW | RW | **PortOverCurrentIndicatorChange**<br>This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the **PortOverCurrentIndicator** bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br>0 = No change in **PortOverCurrentIndicator**<br>1 = **PortOverCurrentIndicator** has changed |
| PRSC | 20 | 0b | RW | RW | **PortResetStatusChange**<br>This bit is set at the end of the 10-ms port reset signal.<br>The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br>0 = Port reset is not complete<br>1 = Port reset is complete |
| Reserved | 31:21 | | | | Reserved |

### *2.2.2 USB Legacy Keyboard Operation*

#### *2.2.2.1 Overview*

To support applications and drivers in non-USB-aware environments (e.g., DOS), the Host Controller needs to provide some amount of hardware support for the emulation of a PS/2 keyboard and/or mouse by their USB equivalents. For Open HCI, this emulation support is provided by a set of registers that are controlled by code running in SMM. Working in conjunction, this hardware and software produces approximately the same behavior-to-application code as would be produced by a PS/2-compatible keyboard and/or mouse interface.

To minimize hardware impact, the Host Controller accesses a USB keyboard and/or mouse using the

standard OpenHCI descriptor-based accesses. The emulation code sets up the appropriate Endpoint Descriptors and Transfer Descriptors that cause data to be sent to or received from a USB keyboard/mouse using the normal USB protocols. When data is received from the keyboard/mouse, the emulation code is notified and becomes responsible for translating the USB keyboard/mouse data into a data sequence that is equivalent to what would be produced by a PS/2-compatible keyboard/mouse interface. The translated data is made available to the system through the legacy keyboard interface I/O addresses at 60h and 64h. Likewise, when data/control is to be sent to the keyboard (as indicated by the system writing to the legacy keyboard interface), the emulation code is notified and becomes responsible for translating the information into appropriate data to be sent to the USB keyboard/mouse through the transfer descriptor mechanism.

On the PS/2 keyboard/mouse interface, a read of I/O port 60h returns the current contents of the keyboard output buffer; a read of I'O port 64h returns the contents of the keyboard status register. An I/O write to port 60h or 64h puts data into the keyboard input buffer (data is being input into the keyboard subsystem). When emulation is enabled, reads and writes of registers 60h and 64h are captured in *HceOutput*, *HceStatus*, and/or *HceInput* operational registers.

The emulation hardware described in this document supports a mixed environment in which either the keyboard or mouse is located on USB, and the other device is attached to a standard PS/2 interface.

#### *2.2.2.2 System Requirements*

The sections below define the system requirements that must be met in order for the OpenHCI legacy support to function properly.

**Host Controller Mapping**

The Host Controller uses memory addresses to enable system software to access its operational registers. In a PCI implementation, the address of the Host Controller operations registers is set in BAR_OHCI. The address range specified in BAR_OHCI must be accessible to SMM code. The address in BAR_OHCI should not be modified by any software while the emulation software has control of the Host Controller. The only exception to this is when the OS is booting and is trying to interrogate the PCI bus. It is common for an OS, as it is loaded, to enumerate and 'size' the various buses on the machine. For a PCI system, the OS typically writes a value to each card's BAR to determine the memory space occupied by that card. If emulation is running during enumeration, the Host Controller may generate an SMI as the OS is changing the BAR from the value that the emulation code is using.

**Intercept Port 60h and 64h Accesses**

When emulation is enabled, I/O accesses of I/O ports 60h and 64h must be handled by the Host Controller. The Host Controller must be positioned in the system so that it can do a positive decode of accesses to I/O addresses 60h and 64h on the PCI bus. If a keyboard controller is present in the system, it must either use subtractive decode or have provisions to disable its decode of ports 60h and 64h. If the legacy keyboard controller uses positive decode and is turned off during emulation, it must be possible for the emulation code to quickly re-enable and disable the legacy keyboard controller's 60h and 64h decode. This is necessary to support a mixed operating environment.

**Interrupts**

The Host Controller must connect to IRQ1 and IRQ12 on the system board and be wired OR with other non-legacy IRQ1 and IRQ12 sources. IRQ1 and IRQ12 from the legacy keyboard controller (if present) must be routed through the Host Controller.

**Run-time Memory**

Legacy emulation requires that the Host Controller have read/write access to a portion of system memory that is not used by a system OS for any purpose. In addition, this memory must be accessible by the host CPU while the host CPU is in SMM.

#### *2.2.2.3 Programming Interface*

The following modification is needed for the HcRevision register:

**Table 2-1 HcRevision Register**

| HcRevision - 32 bits | | | | | |
|---|---|---|---|---|---|
| **Field Name** | **Bits** | **Reset** | **HCD** | **HC** | **Description** |
| Revision | 7:0 | 10h | R | R | This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 11h corresponds to version 1.1. All of the HC implementations that are compliant with this specification will have a value of 10h. |
| Legacy | 8 | 1b | R | R | This read-only field is 1 to indicate that the legacy support registers are present in this HC. |
| Reserved | 31:9 | | | | Reserved |

**Legacy Support Registers**

Four operational registers are used to provide the legacy support. Each of these registers is located on a 32-bit boundary. The offset of these registers is relative to the base address of the Host Controller operational registers with HceControl located at offset 100h.

**Table 2-2 Legacy Support Registers**

| Offset | Register | Description |
|---|---|---|
| 100h | HceControl | Used to enable and control the emulation hardware and report various status information. |
| 104h | HceInput | The emulation side of the legacy Input Buffer register. |
| 108h | HceOutput | The emulation side of the legacy Output Buffer register where the keyboard and mouse data is to be written by software. |
| 10Ch | HceStatus | The emulation side of the legacy Status register. |

Three of the operational registers (HceStatus, HceInput, HceOutput) are accessible at I/O address 60h and 64h when emulation is enabled. Reads and writes to the registers using I/O addresses have side effects as outlined in *Table 2-3*.

**Table 2-3 Emulated Registers**

| I/O Address | Cycle Type | Register Contents Accessed/Modified | Side Effects |
|---|---|---|---|
| 60h | IN | HceOutput | IN from port 60h will set OutputFull in HceStatus to 0 |
| 60h | OUT | HceInput | OUT to port 60h will set InputFull to 1 and CmdData to 0 in HceStatus. |
| 64h | IN | HceStatus | IN from port 64h returns current value of HceStatus with no other side effect. |
| 64h | OUT | HceInput | OUT to port 64h will set InputFull to 0 and CmdData in HceStatus to 1. |

**HceInput Register**

**Table 2-4 HceInput Registers**

| HceInput – RW - 32 bits | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| InputData | 7:0 | 00h | This register holds data that is written to I/O ports 60h and 64h. |
| Reserved | 31:8 | | Reserved |

I/O data that is written to ports 60h and 64h is captured in this register when emulation is enabled. This register may be read or written directly by accessing it with its memory address in the Host Controller's operational register space. When accessed directly with a memory cycle, reads and writes of this register have no side effects.

**HceOutput Register**

**Table 2-5 HceOutput Register**

| HceOutput – RW - 32 bits | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| OutputData | 7:0 | 00h | This register hosts data that is returned when an I/O read of port 60h is performed by application software. |
| Reserved | 31:8 | | Reserved |

The data placed in this register by the emulation software is returned when I/O port 60h is read and emulation is enabled. On a read of this location, the **OutputFull** bit in HceStatus is set to 0.

**HceStatus Register**

**Table 2-6 HceStatus Register**

| HceStatus – RW - 32 bits | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| OutputFull | 0 | 0b | The HC sets this bit to 0 on a read of I/O port 60h. If IRQEn is set and AuxOutputFull is set to 0, then an IRQ1 is generated as long as this bit is set to 1. If IRQEn is set and AuxOutputFull is set to 1, then an IRQ12 is generated as long as this bit is set to 1. While this bit is 0 and CharacterPending in HceControl is set to 1, an emulation interrupt condition exists. |
| InputFull | 1 | 0b | Except for the case of a Gate A20 sequence, this bit is set to 1 on an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists. |
| Flag | 2 | 0b | Nominally used as a system flag by software to indicate a warm or cold boot. |
| CmdData | 3 | 0b | The HC sets this bit to 0 on an I/O write to port 60h and to 1 on an I/O write to port 64h. |
| Inhibit Switch | 4 | 0b | This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited. |
| AuxOutputFull | 5 | 0b | IRQ12 is asserted whenever this bit is set to 1 and OutputFull is set to 1 and the IRQEn bit is set. |
| Time-out | 6 | 0b | Used to indicate a time-out |
| Parity | 7 | 0b | Indicates parity error on keyboard/mouse data. |
| Reserved | 31:8 | | Reserved |

The contents of the HceStatus Register are returned on an I/O Read of port 64h when emulation is enabled. Reads and writes of port 60h and writes to port 64h can cause changes in this register. Emulation software can directly access this register through its memory address in the Host Controller's operational register space. Accessing this register through its memory address produces no side effects.

**HceControl Register**

**Table 2-7 HceControl Register**

| HceControl - 32 bits | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Reset** | **Description** |
| EmulationEnable | 0 | 0b | When set to 1, the HC is enabled for legacy emulation. The HC decodes accesses to I/O registers 60h and 64h and generates IRQ1 and/or IRQ12 when appropriate. Additionally, the HC generate s an emulation interrupt at appropriate times to invoke the emulation software. |
| EmulationInterrupt | 1 | - | This bit is a static decode of the emulation interrupt condition. [Read-only] |
| CharacterPending | 2 | 0b | When set, an emulation interrupt is generated when the *OutputFull* bit of the *HceStatus* register is set to 0. |
| IRQEn | 3 | 0b | When set, the HC generates IRQ1 or IRQ12 as long as the *OutputFull* bit in *HceStatus* is set to 1. If the *AuxOutputFull* bit of *HceStatus* is 0, then IRQ1 is generated; if it is 1, then an IRQ12 is generated. |
| ExternalIRQEn | 4 | 0b | When set to 1, IRQ1 and IRQ12 from the keyboard controller causes an emulation interrupt. The function controlled by this bit is independent of the setting of the *EmulationEnable* bit in this register. |
| GateA20Sequence | 5 | 0b | Set by HC when a data value of D1h is written to I/O port 64h. Cleared by HC on write to I/O port 64h of any value other than D1h. |
| IRQ1Active | 6 | 0b | Indicates that a positive transition on IRQ1 from keyboard controller has occurred. SW may write a 1 to this bit to clear it (set it to 0). SW write of a 0 to this bit has no effect. |
| IRQ12Active | 7 | 0b | Indicates that a positive transition on IRQ12 from keyboard controller has occurred. SW may write a 1 to this bit to clear it (set it to 0). SW write of a 0 to this bit has no effect. |
| A20State | 8 | 0b | Indicates current state of Gate A20 on keyboard controller. Used to compare against value written to 60h when *GateA20Sequence* is active. |
| Reserved | 31:9 | - | Must read as 0s. |

### *2.2.3 EHCI Registers (Device 19, Function 5)*

The Enhanced USB Host Controller contains two sets of software accessible hardware registers—Memory-mapped Host Controller Registers and optional PCI configuration registers (PCI_Reg).

Mapping into non-cacheable memory, Memory-mapped USB Host Controller Registers consists of a set of read-only Capability registers (MEM_Reg) , a set of read/write operational registers(EOR_Reg) and a set of read/write Debug Port registers (DBUG_Reg). Implemented as memory-mapped I/O space, the operational registers are 32 bits in length and should be read and written as Dwords.

#### *2.2.3.1 PCI Configuration Registers*

| Registers Name | Offset Address |
|---|---|
| Device / Vendor ID | 00h |
| Command | 04h |
| Status | 06h |
| Revision ID / Class Code | 08h |
| Miscellaneous | 0Ch |

| | |
|---|---|
| Base Address – BAR_EHCI | 10h |
| Subsystem ID / Subsystem Vendor ID | 2Ch |
| Capability Pointer | 34h |
| Interrupt Line | 3Ch |
| EHCI Misc Control | 50h |
| Serial Bus Release Number – SBRN | 60h |
| Frame Length Adjustment – FLADJ | 61h |
| PME Control | C0h |
| PME Data / Status | C4h |
| MSI Control | D0h |
| MSI Address | D4h |
| MSI Data | D8h |
| EHCI Debug Port Support | E4h |
| USB Legacy Support Extended Capability – USBLEGSUP | EECP+0h[1] |
| USB Legacy Support Control/Status - USBLEGCTLSTS | EECP+4h[1] |
| [1]The EECP field is in the read-only HCCPARAMS register [MEM_Reg: 08h] with the value of A0h. | |

| DEVICE / VENDOR ID – R - 32 bits - [PCI_Reg : 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VEND_ID | 15:0 | 1002h | Vendor ID |
| DEV_ID | 31:16 | Function 5: 4386h | Device ID |

| Command – RW - 16 bits - [PCI_Reg : 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Space Accesses | 0 | 0b | A value of 0 disables the device response.<br>A value of 1 allows the device to respond to I/O Space accesses. |
| Memory Space Accesses | 1 | 0b | A value of 0 disables the device response.<br>A value of 1 allows the device to respond to Memory Space accesses. |
| Bus Master | 2 | 0b | A value of 0 disables the device from generating PCI accesses.<br>A value of 1 allows the device to behave as a bus master. |
| Special Cycle | 3 | 0b | Hard-wired to 0, indicating no Special Cycle support. |
| Memory Write and Invalidate Command | 4 | 0b | When it is 0, Memory Write must be used.<br>When it is 1, masters may generate the command. |
| VGA Palette Register Accesses | 5 | 0b | Hard-wired to 0, indicating the device should treat palette write accesses like all other accesses. |
| Parity Enable | 6 | 0b | When it is 1, the device must take its normal action when a parity error is detected.<br>When it is 0, the device sets its Detected Parity Error status bit (bit 15 in the Status register) when an error is detected, but does not assert PERR# and continues normal operation. |
| Reserved | 7 | 0b | Hard-wired to 0 per PCI2.3 spec. |
| SERR# Enable | 8 | 0b | A value of 0 disables the SERR# driver.<br>A value of 1 enables the SERR# driver.<br>Address parity errors are reported only if this bit and bit [6] are 1. |
| Fast Back-to-Back Enable | 9 | 0b | A value of 0 means fast back-to-back transactions to the same agent only are allowed.<br>A value of 1 means the master is allowed to generate fast back-to-back transactions to different agents. |
| Interrupt Disable | 10 | 0b | A value of 0 enables the assertion of the device/function's INTx# signal.<br>A value of 1 disables the assertion of the device/function's INTx# signal. |
| Reserved | 15:11 | | Reserved |

| Status – R - 16 bits - [PCI_Reg : 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | | Reserved |
| Interrupt Status | 3 | 0b | This bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit. |
| Capabilities List | 4 | 1b | A value of 0 indicates that no New Capabilities linked list is available.<br>A value of 1 indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 66 MHz Capable | 5 | 1b | Hard-wired to 1, indicating 66MHz capable. |
| Reserved | 6 | | Reserved |
| Fast Back-to-Back Capable | 7 | 1b | Hard-wired to 1, indicating Fast Back-to-Back capable. |
| Master Data Parity Error | 8 | 0b | This bit is set only when three conditions are met: 1) the bus agent asserted PERR# itself (on a read) or observed PERR# asserted (on a write); 2) the agent setting the bit acted as the bus master for the operation in which the error occurred; and 3) the Parity Error Response bit (Command register) is set. |
| DEVSEL timing | 10:9 | 01b | Hard-wired to 01b – medium timing |
| Signaled Target Abort | 11 | 0b | This bit is set by a target device whenever it terminates a transaction with Target-Abort. |
| Received Target Abort | 12 | 0b | This bit is set by a master device whenever its transaction is terminated with Target-Abort. |
| Received Master Abort | 13 | 0b | This bit is set by a master device whenever its transaction (except for Special Cycle) is terminated with Master-Abort. |
| Signaled System Error | 14 | 0b | This bit is set whenever the device asserts SERR#. |
| Detected Parity Error | 15 | 0b | This bit is set by the device whenever it detects a parity error, even if parity error handling is disabled (as controlled by bit 6 in the Command register). |

| Revision ID / Class Code – R - 32 bits - [PCI_Reg : 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | Revision ID. |
| PI | 15:8 | 20h | Programming Interface. A constant value of '20h' indentifies the device being an EHCI Host Controller. |
| SC | 23:16 | 03h | Sub Class. A constant value of '03h' indentifies the device being of Universal Serial Bus. |
| BC | 31:24 | 0Ch | Base Class. A constant value of '0Ch' identifies the device being a Serial Bus Controller. |

| Miscellaneous – RW - 32 bits - [PCI_Reg : 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | This read/write field specifies the system cacheline size in units of DWORDs and must be initialized to 00h. |
| Latency Timer | 15:8 | 00h | [9:8] hard-wired to 00b, resulting in a timer granularity of at least four clocks. This field specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. |

| Miscellaneous – RW - 32 bits - [PCI_Reg : 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 23:16 | 00h | This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space) and also whether or not the device contains multiple functions.<br>EHCI has single function and bit[23:16] hard-wired to 00h.<br>Read Only. |
| BIST | 31:24 | 00h | Hard-wired to 00h, indicating no build-in BIST support. |

| BAR_EHCI – RW - 32 bits - [PCI_Reg : 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IND | 0 | 0b | Indicator. A constant value of '0' indicates that the operational registers of the device are mapped into memory space of the main memory of the PC host system.<br>Read Only. |
| TP | 2:1 | 0h | Type. A constant value of '00b' indicates that the base register is 32-bit wide and can be placed anywhere in the 32-bit memory space; i.e., lower 4 GB of the main memory of the PC host.<br>Read Only. |
| PM | 3 | 0b | Prefetch Memory. A constant value of '0' indicates that there is no support for "prefetchable memory".<br>Read Only. |
| Reserved | 7:4 | 0h | Read Only. |
| BA | 31:8 | 0h | Base Address. Corresponds to memory address signals [31:8]. |
| BAR register. Base address used for the memory mapped capability and operational registers. | | | |

| Subsystem ID / Subsystem Vendor ID – RW - 32 bits - [PCI_Reg : 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Can only be written once by software. |
| Subsystem ID | 31:16 | 0h | Can only be written once by software. |

| Capability Pointer – R - 8 bits - [PCI_Reg : 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability Pointer | 7:0 | C0h | Address of the 1st element of capability link. |

| Interrupt Line - RW - 32 bits - [PCI_Reg : 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | The Interrupt Line is a field used to communicate interrupt line routing information. The register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST software will write the routing information into this register as it initializes and configures the system.<br><br>The value in this field tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. The device itself does not use this value; rather it is used by device drivers and operating systems. Device drivers and operating systems can use this information to determine priority and vector information. Values in this register are system architecture specific. |
| Interrupt Pin | 15:8 | 04h | Read Only by default.<br>Hard-wired to 04h, which corresponds to using INTD#. |
| MIN_GNT | 23:16 | 00h | Read Only. Hard-wired to 00h to indicate no major requirements for the settings of Latency Timers. |
| MAX_LAT | 31:24 | 00h | Read Only. Hard-wired to 00h to indicate no major requirements for the settings of Latency Timers. |

| EHCI Misc Control – RW - 32 bits - [PCI_Reg : 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 4:0 | 00h | Reserved |
| PME Disable | 5 | 0b | Set to 1 to disable EHCI PME support |
| MSI Disable | 6 | 0b | Set to 1 to disable EHCI MSI support |
| Reserved | 15:7 | 000h | Reserved |
| Cache Timer Control | 19:16 | Eh | Control the purge timeout timer if HC doesn't come back to request the data. |
| Cache Prefetch Disable | 20 | 0b | 0: Enable EHCI cache prefetch.<br>1: Disable EHCI cache prefetch. |
| Reserved | 23:21 | | |
| Disable Async QH Cache on IN xfer | 24 | 0b | Set to 1 to disable async QH/QTD cache during IN xfer. |

| Counter | Max Time (ns) | Min Time (ns) |
|---|---|---|
| 0 | 45 | 30 |
| 1 | 90 | 60 |
| 2 | 180 | 120 |
| 3 | 360 | 240 |
| 4 | 720 | 480 |
| 5 | 1440 | 960 |
| 6 | 2880 | 1920 |
| 7 | 5760 | 3840 |
| 8 | 11520 | 7680 |
| 9 | 23040 | 15360 |
| A | 46080 | 30720 |
| B | 92160 | 61440 |
| C | 184320 | 122880 |
| D | 368640 | 245760 |
| E | 737280 | 491520 |
| F | No limit | |

| EHCI Misc Control – RW - 32 bits - [PCI_Reg : 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Disable Async QH Cache on OUT xfer | 25 | 0b | Set to 1 to disable async QH/QTD cache during OUT xfer. |
| Disable Async Data Cache | 26 | 0b | Set to 1 to disable async data cache request. |
| Disable Periodic List Cache | 27 | 0b | Set to 1 to disable periodic list cache. |
| Reserved | 31:28 | 0h | Reserved |

| SBRN – R - 8 bits - [PCI_Reg : 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SBRN | 7:0 | 20h | Hard-wired to 20h. |

| FLADJ – RW - 8 bits - [PCI_Reg : 61h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FLADJ | 5:0 | 20h | Frame Length Timing Value. Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. |
| Reserved | 7:6 | | Reserved. |

| FLADJ Value in decimals [hexadecimal value] | Frame Length (# High Speed bit times in decimals) |
|---|---|
| 0 [00h] | 59488 |
| 1 [01h] | 59504 |
| 2 [02h] | 59520 |
| … | … |
| 31 [1Fh] | 59984 |
| 32 [20h] | 60000 |
| … | … |
| 62 [3Eh] | 60480 |
| 63 [3Fh] | 60496 |

| PME Control – RW - 32 bits - [PCI_Reg : C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cap_ID | 7:0 | 01h | Read only.<br>A value of “01h” identifies the linked list item as being the PCI Power Management registers. |
| Next ItemPointer | 15:8 | D0h | Read only.<br>This field provides an offset into the function’s PCI Configuration Space pointing to the location of next item in the function’s capability list. If there are no additional items in the Capabilities List, this register is set to 00h. |
| Version | 18:16 | 010b | Read only.<br>A value of “010b” indicates that this function complies with Revision 1.1 of the PCI Power Management Interface Specification. |
| PME clock | 19 | 0b | Read only.<br>When this bit is a “0”, it indicates that no PCI clock is required for the function to generate PME#. |
| Reserved | 20 | | Reserved |

| PME Control – RW - 32 bits - [PCI_Reg : C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DSI | 21 | 0b | Read only.<br>The Device Specific Initialization bit indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. |
| Aux_Current | 24:22 | 000b | Read only.<br>This 3 bit field reports the 3.3Vaux auxiliary current requirements for the PCI function.<br>If the Data Register has been implemented by this function:<br>• Reads of this field must return a value of "000b".<br>• The Data Register takes precedence over this field for 3.3Vaux current requirement reporting. |
| D1_Support | 25 | 1b | If this bit is a "1", this function supports the D1 Power Management State. |
| D2_Support | 26 | 1b | If this bit is a "1", this function supports the D2 Power Management State. |
| PME_Support | 31:27 | 0Fh | Read only.<br>This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state.<br>bit(31) 1XXXXb - PME# can be asserted from D3cold bit(30) X1XXXb - PME# can be asserted from $D3_{hot}$<br>bit(29) XX1XXb - PME# can be asserted from D2<br>bit(28) XXX1Xb - PME# can be asserted from D1<br>bit(27) XXXX1b - PME# can be asserted from D0 |

| PME Data / Status – RW - 32 bits - [PCI_Reg : C4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PowerState | 1:0 | 00b | This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the field values is given below.<br>00b - D0<br>01b - D1<br>10b - D2<br>11b - $D3_{hot}$<br>If software attempts to write an unsupported, optional state to this field, the write operation must be completed normally on the bus; however, the data is discarded and no state change occurs. |
| Reserved | 7:2 | | Reserved |
| PME_En | 8 | 0b | A "1" enables the function to assert PME#. When "0", PME# assertion is disabled. This bit defaults to "0" if the function does not support PME# generation from $D3_{cold}$. |
| Data_Select | 12:9 | 0000b | This 4-bit field is used to select which data is to be reported through the Data register and Data_Scale field. |
| Data_Scale | 14:13 | 00b | This 2-bit read-only field indicates the scaling factor to be used when interpreting the value of the Data register. The value and meaning of this field will vary depending on which data value has been selected by the Data_Select field. |
| PME_Status | 15 | 0b | This bit is set when the function would normally assert the PME# signal independent of the state of the PME_En bit. |
| Reserved | 21:16 | | Reserved |

| PME Data / Status – RW - 32 bits - [PCI_Reg : C4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| B2_B3# | 22 | 1b | Read only.<br>The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. A "1" indicates that when the bridge function is programmed to $D3_{hot}$, its secondary bus's PCI clock will be stopped (B2). |
| BPCC_En | 23 | 0b | Read only.<br>A "0" indicates that the bus power/clock control policies are disabled. When the Bus Power/Clock Control mechanism is disabled, the bridge's PMCSR PowerState field cannot be used by the system software to control the power or clock of the bridge's secondary bus. |
| Data | 31:24 | 00h | Read only.<br>This register is used to report the state dependent data requested by the Data_Select field. The value of this register is scaled by the value reported by the Data_Scale field. |

| MSI Control – RW - 32 bits - [PCI_Reg : D0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI USB | 7:0 | 05h | MSI USB ID. Read only. |
| Next Item Pointer | 15:8 | E4h | Pointer to next capability structure |
| MSI Control Out | 16 | 0b | Set to 1 to disable IRQ. Use MSI instead. |
| Reserved | 19:17 | 0h | Reserved |
| MSI Control | 22:20 | 0h | MSI control field |
| 64-bit Address Capable | 23 | 0b | If EHCI is in 64 bit address mode as specified by 64-bit Addressing Capability bit in HCCPARAMS register [MEM Reg: 08h] , this bit is set to 1 indicating that EHCI is capable of generating a 64-bit message address. Otherwise it is set to 0 indicating the EHCI is not capable of generating a 64-bit address.<br>Read only |
| Reserved | 31:24 | 00h | Reserved |

| MSI Address – RW - 32 bits - [PCI_Reg : D4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Address | 31:0 | 0h | System-specified message address. |

| MSI Data – RW - 16 bits - [PCI_Reg : D8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Data | 15:0 | 0h | System-specified message |

| DBUG_PRT Control – R - 32 bits - [PCI_Reg : E4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CAP_ID | 7:0 | 0Ah | The value of 0Ah in this field identifies that the function supports a Debug Port. |
| Next Item Pointer | 15:8 | 00h | Pointer to next capability structure |
| Offset | 28:16 | 0E0h | This 12 bit field indicates the byte offset (up to 4K) within the BAR indicated by *BAR#*. This offset is required to be DWORD aligned and therefore bits 16 and 17 are always zero. |

| DBUG_PRT Control – R - 32 bits - [PCI_Reg : E4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Bar # | 31:29 | 1h | A 3-bit field, which indicates which one of the possible 6 Base Address Register offsets, contains the Debug Port registers. For example, a value of 1h indicates the first BAR (offset 10h) while a value of 5 indicates that the BAR at 20h. This offset is independent as to whether the BAR is 32 or 64 bit. For example, if the offset were 3 indicating that the BAR at offset 18h contains the Debug Port. BARs at offset 10 and 14h may or may not be implemented. This field is read only and only values 1-6h are valid. (A 64-bit BAR is allowed.) Only a memory BAR is allowed. |

| USBLEGSUP – RW - 32 bits - [PCI_Reg : EECP + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 01h | This field identifies the extended capability. A value of 01h identifies the capability as Legacy Support. This extended capability requires one additional 32-bit register for control/status information, and this register is located at offset EECP+04h.<br>Read Only. |
| Next EHCI Extended Capability Pointer | 15:8 | 00h | This field points to the PCI configuration space offset of the next extended capability pointer. A value of 00h indicates the end of the extended capability list.<br>Read Only. |
| HC BIOS Owned Semaphore | 16 | 0b | The BIOS sets this bit to establish ownership of the EHCI controller. System BIOS will set this bit to a zero in response to a request for ownership of the EHCI controller by system software. |
| Reserved | 23:17 | | These bits are reserved and must be set to zero. |
| HC OS Owned Semaphore | 24 | 0b | System software sets this bit to request ownership of the EHCI controller. Ownership is obtained when this bit reads as one and the HC BIOS Owned Semaphore bit reads as 0. |
| Reserved | 31:25 | | These bits are reserved and must be set to zero. |

| USBLEGCTLSTS – RW - 32 bits - [PCI_Reg : EECP + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| USB SMI Enable | 0 | 0b | When this bit is a one, and the *SMI on USB Complete* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| SMI on USB Error Enable | 1 | 0b | When this bit is a one, and the *SMI on USB Error* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| SMI on Port Change Enable | 2 | 0b | When this bit is a one, and the *SMI on Port Change Detect* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| SMI on Frame List Rollover Enable R/W | 3 | 0b | When this bit is a one, and the *SMI on Frame List Rollover* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| SMI on Host System Error Enable | 4 | 0b | When this bit is a one, and the *SMI on Host System Error* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| SMI on Async Advance Enable | 5 | 0b | When this bit is a one, and the *SMI on Async Advance* bit (above) in this register is a one, the host controller will issue an SMI immediately. |
| Reserved. | 12:6 | | These bits are reserved and must be set to zero. |

| USBLEGCTLSTS – RW - 32 bits - [PCI_Reg : EECP + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMI on OS Ownership Enable | 13 | 0b | When this bit is a one AND the *OS Ownership Change* bit is one, the host controller will issue an SMI. |
| SMI on PCI Command Enable | 14 | 0b | When this bit is one and *SMI on PCI Command* is one, then the host controller will issue an SMI. |
| SMI on BAR Enable | 15 | 0b | When this bit is one and *SMI on BAR* is one, then the host controller will issue an SMI. |
| SMI on USB Complete | 16 | 0b | Shadow bit of *USB Interrupt* (USBINT) bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *USB Interrupt* bit in the USBSTS register. Read Only. |
| SMI on USB Error | 17 | 0b | Shadow bit of *USB Error Interrupt* (USBERRINT) bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *USB Error Interrupt* bit in the USBSTS register. Read Only. |
| SMI on Port Change Detect. | 18 | 0b | Shadow bit of *Port Change Detect* bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *Port Change Detect* bit in the USBSTS register. Read Only. |
| SMI on Frame List Rollover | 19 | 0b | Shadow bit of *Frame List Rollover* bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *Frame List Rollover* bit in the USBSTS register. Read Only. |
| SMI on Host System Error | 20 | 0b | Shadow bit of *Host System Error* bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *Host System Error* bit in the USBSTS register. Read Only. |
| SMI on Async Advance | 21 | 0b | Shadow bit of the *Interrupt on Async Advance* bit in the USBSTS register. To set this bit to a zero, system software must write a one to the *Interrupt on Async Advance* bit in the USBSTS register. Read Only. |
| Reserved. | 28:22 | | These bits are reserved and must be set to zero. |
| SMI on OS Ownership Change | 29 | 0b | This bit is set to one whenever the *HC OS Owned Semaphore* bit in the USBLEGSUP register transitions from 1 to 0 or 0 to 1. |
| SMI on PCI Command | 30 | 0b | This bit is set to one whenever the *PCI Command Register* is written. |
| SMI on BAR R/WC | 31 | 0b | This bit is set to one whenever the *Base Address Register* (BAR) is written. |

#### *2.2.3.2 Host Controller Capability Registers (MEM_Reg)*

This block of registers is memory-mapped. Access address is equal to offset address plus base address defined in BAR[PCI_Reg : 10h].

| Registers Name | Offset Address |
|---|---|
| Capability Register Length - CAPLENGTH | 00h |
| Reserved | 01h |
| Host Controller Interface Version – HCIVERSION | 02h |
| Structural Parameters – HCSPARAMS | 04h |
| Capability Parameters - HCCPARMAS | 08h |
| Companion Port Route Description – HCSP-PORTROUTE | 0Ch |

| CAPLENGTH – R - 8 bits - [MEM_Reg : 00h] |
|---|
| **Description** |
| This register is used as an offset to add to register base to find the beginning of the Operational Register Space. Default value = 20h. |

| HCIVERSION – R - 16 bits - [MEM_Reg : 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| HCIVERSION | 15:0 | 0100h | This is a two-byte register containing a BCD encoding of the version number of interface to which this host controller interface conforms. |

| HCSPARAMS – R - 32 bits - [MEM_Reg : 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| N_PORTS | 3:0 | Ah | This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1H to FH. A zero in this field is undefined. |
| Port Power Control (PPC) | 4 | 0b | This field indicates whether the host controller implementation includes port power control. A one in this bit indicates the ports have port power switches. A zero in this bit indicates the port does not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register. |
| Reserved | 6:5 | | These bits are reserved and should be set to zero. |
| Port Routing Rules | 7 | 0b | This field indicates the method used by this implementation for how all ports are mapped to companion controllers. The value of this field has the following interpretation:<br>0 = The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on.<br>1 = The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTROUTE array. |
| Number of Ports per Companion Controller (N_PCC) | 11:8 | 2h | This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC. |
| Number of Companion Controller (N_CC) | 15:12 | 5h | This field indicates the number of companion controllers associated with this USB 2.0 host controller. A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports. A value larger than zero in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports. |
| Port Indicators (P_INDICATOR) | 16 | 0b | This bit indicates whether the ports support port indicator control. When this bit is a one, the port status and control registers include a read/writeable field for controlling the state of the port indicator. |

| HCSPARAMS – R - 32 bits - [MEM_Reg : 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 19:17 | | These bits are reserved and should be set to zero. |
| Debug Port Number | 23:20 | 1h | Optional. This register identifies which of the host controller ports is the debug port. The value is the port number (one-based) of the debug port. A non-zero value in this field indicates the presence of a debug port. The value in this register must not be greater than N_PORTS. |
| Reserved | 31:24 | | These bits are reserved and should be set to zero. |

| HCCPARAMS – R - 32 bits - [MEM_Reg : 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| 64-bit Addressing Capability | 0 | 0b | This field documents the addressing range capability of this implementation.<br>0 = Data structures using 32-bit address memory pointers<br>1 = Data structures using 64-bit address memory pointers |
| Programmable Frame List Flag | 1 | 1b | If this bit is set to a zero, then system software must use a frame list length of 1024 elements with this host controller. The USBCMD register *Frame List Size* field is a read-only register and should be set to zero. If set to a one, then system software can specify and use a smaller frame list and configure the host controller via the USBCMD register *Frame List Size* field. The frame list must always be aligned on a 4K page boundary. This requirement ensures that the frame list is always physically contiguous. |
| Asynchronous Schedule Park Capability | 2 | 0b | If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USBCMD register. |
| Reserved | 3 | | These bits are reserved and should be set to zero. |
| Isochronous Scheduling Threshold | 7:4 | 1h | This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame. |
| EHCI Extended Capabilities Pointer (EECP) | 15:8 | A0h | This optional field indicates the existence of a capabilities list. A value of 00h indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capability. The pointer value must be 40h or greater if implemented to maintain the consistency of the PCI header defined for this class of device. |
| Reserved | 31:16 | | These bits are reserved and should be set to zero. |

| HCSP-PORTROUTE – R - 60 bits - [MEM_Reg : 0Ch] |
|---|
| **Description** |
| This optional field is valid only if *Port Routing Rules* field in the HCSPARAMS register is set to a one. This field is a 15-element nibble array (each 4 bits is one array element). Each array location corresponds one-to-one with a physical port provided by the host controller. |

#### *2.2.3.3 Host Controller Operational Registers (EOR_Reg)*

This block of registers is memory-mapped. The base offset, EHCI_EOR, is defined in CAPLENGTH register (MEM_Reg: 00h, default value = 20h).

| Registers Name | Offset Address |
|---|---|
| USB Command – USBCMD | EHCI_EOR + 00h |
| USB Satus – USBSTS | EHCI_EOR + 04h |
| USB Interrupt Enable – USBINTR | EHCI_EOR + 08h |
| USB Frame Index – FRINDEX | EHCI_EOR + 0Ch |
| 4G Segment Selector – CTRLDSSEGMENT | EHCI_EOR + 10h |
| Frame List Base Address – PERIODICLISTBASE | EHCI_EOR + 14h |
| Next Asynchronous List Address – ASYNCLISTADDR | EHCI_EOR + 18h |
| Reserved | EHCI_EOR + (1Ch~3Fh) |
| Configured Flag – CONFIGFLAG | EHCI_EOR + 40h |
| Port Status/Control – PORTSC (1-N_PORTS) | EHCI_EOR + (44h~68h) |
| Packet Buffer Threshold Values | EHCI_EOR + 84h |
| USB PHY Status 0 | EHCI_EOR + 88h |
| USB PHY Status 1 | EHCI_EOR + 8Ch |
| USB PHY Status 2 | EHCI_EOR + 90h |
| UTMI Control | EHCI_EOR + 94h |
| Bist Control / Loopback Test | EHCI_EOR + 98h |
| EOR MISC Control | EHCI_EOR + 9Ch |
| USB Phy Calibration | EHCI_EOR + A0h |
| EOR Debug Purpose | EHCI_EOR + A8h |
| USB Debug Port | 0E0h~0F0h (* Note) |
| The base offset of Debug Port registers is defined directly in DBUG_PRT Control register (EHCI_PCI_CFG xE4[28:16]), regardless of the value in CAPLENGTH register (MEM_Reg: 00h) so range is equivalent to EHCI_EOR + (C0h~D0h). | |

| USBCMD – RW - 32 bits - [EOR_Reg : EHCI_EOR + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Run/Stop (RS) | 0 | 0b | 1=Run, 0=Stop.<br>When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the host controller is in the Halted state (i.e. *HCHalted* in the USBSTS register is a one). Doing so will yield undefined results. |

| USBCMD – RW - 32 bits - [EOR_Reg : EHCI_EOR + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Host Controller Reset (HCRESET) | 1 | 0b | This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset. When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. PCI Configuration registers are not affected by this reset. All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s). Software must reinitialize the host controller in order to return the host controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the *HCHalted* bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. |
| Frame List Size | 3:2 | 00b | This field is R/W only if *Programmable Frame List Flag* in the HCCPARAMS registers is set to a one. This field specifies the size of the frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. Values mean:<br>00b = 1024 elements (4096 bytes) Default value<br>01b = 512 elements (2048 bytes)<br>10b = 256 elements (1024 bytes) – for resource-constrained environments<br>11b = Reserved<br>[Read/Write or Read-only] |
| Periodic Schedule Enable | 4 | 0b | This bit controls whether the host controller skips processing the Periodic Schedule.<br>0 = Do not process the Periodic Schedule<br>1 = Use the PERIODICLISTBASE register to access the Periodic Schedule. |
| Asynchronous Schedule Enable | 5 | 0b | This bit controls whether the host controller skips processing the Asynchronous Schedule.<br>0b = Do not process the Asynchronous Schedule<br>1b = Use the ASYNCLISTADDR register to access the Asynchronous Schedule. |
| Interrupt on Async Advance Doorbell | 6 | 0b | This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to *ring* the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the *Interrupt on Async Advance* status bit in the USBSTS register. If the *Interrupt on Async Advance Enable* bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to a zero after it has set the *Interrupt on Async Advance* status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results. |
| Light Host Controller Reset (Optional) | 7 | 0b | This control bit is not required. If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers. For example, the PORSTC registers should not be reset to their default values and the CF bit setting should not go to zero (retaining port ownership relationships). A host software read of this bit as zero indicates the Light Host Controller Reset has completed and it is safe for host software to re-initialize the host controller. A host software read of this bit as a one indicates the Light Host Controller Reset has not yet completed.<br>If not implemented a read of this field will always return a zero. |

| USBCMD – RW - 32 bits - [EOR_Reg : EHCI_EOR + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Asynchronous Schedule Park Mode Count (Optional) | 9:8 | 00b | If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is RO. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in undefined behavior.<br>[Read/Write or Read-only] |
| Reserved | 10 | | This bit is reserved and should be set to Zero. |
| Asynchronous Schedule Park Mode Enable (Optional) | 11 | 0b | [Read-only]<br>If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. |
| Reserved | 15:12 | | This bit is reserved and should be set to Zero. |
| Interrupt Threshold Control | 23:16 | 08h | This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.<br><br>00h = Reserved<br>01h = 1 micro-frame<br>02h = 2 micro-frames<br>04h = 4 micro-frames<br>08h = 8 micro-frames (default, equates to 1 ms)<br>10h = 16 micro-frames (2 ms)<br>20h = 32 micro-frames (4 ms)<br>40h = 64 micro-frames (8 ms)<br><br>Any other value in this register yields undefined results. Software modifications to this bit while HCHalted bit is equal to zero results in undefined behavior. |
| Reserved | 31:24 | | These bits are reserved and should be set to Zeros. |

| USBSTS - RW - 32 bits - [EOR_Reg : EHCI_EOR + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| USBINT | 0 | 0b | USB Interrupt. The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes). |
| USBERRINT | 1 | 0b | USB Error Interrupt . The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and *USBINT* bit are set. |

| USBSTS - RW - 32 bits - [EOR_Reg : EHCI_EOR + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port Change Detect | 2 | 0b | Port Change Detect. The Host Controller sets this bit to a one when any port for which the *Port Owner* bit is set to zero (see Section 2.3.9) has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Change being set to a one after system software has relinquished ownership of a connected port by writing a zero to a port's *Port Owner* bit. This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force port resume, over-current change, enable/disable change and connect status change). |
| Frame List Rollover | 3 | 0b | Frame List Rollover. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the Frame List Size field of the USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX[12] toggles. |
| Host System Error | 4 | 0b | Host System Error. The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Master Abort, and PCI Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. |
| Interrupt on Async Advance | 5 | 0b | Interrupt on Async Advance. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the *Interrupt on Async Advance Doorbell* bit in the USBCMD register. This status bit indicates the assertion of that interrupt source. |
| Reserved | 11:6 | | These bits are reserved and should be set to zero. |
| HCHalted | 12 | 1b | HCHalted. This bit is a zero whenever the *Run/Stop* bit is a one. The Host Controller sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error). [Read-only] |
| Reclamation | 13 | 0b | Reclamation. This is a read-only status bit, which is used to detect an empty asynchronous schedule. [Read-only] |
| Periodic Schedule Status | 14 | 0b | Periodic Schedule Status. The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). [Read-only] |
| Asynchronous Schedule Status | 15 | 0b | Asynchronous Schedule Status. The bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit in the USBCMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). [Read-only] |
| Reserved | 31:16 | | These bits are reserved and should be set to zero. |

| USBINTR –RW - 32 bits - [EOR_Reg : EHCI_EOR + 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| USB Interrupt Enable | 0 | 0b | When this bit is a one, and the *USBINT* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the *USBINT* bit. |
| USB Error Interrupt Enable | 1 | 0b | When this bit is a one, and the *USBERRINT* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBERRINT bit. |
| Port Change Interrupt Enable | 2 | 0b | When this bit is a one, and the *Port Change Detect* bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Change Detect bit. |
| Frame List Rollover Enable | 3 | 0b | When this bit is a one, and the *Frame List Rollover* bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit. |
| Host System Error Enable | 4 | 0b | When this bit is a one, and the *Host System Error Status* bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit. |
| Interrupt on Async Advance Enable | 5 | 0b | When this bit is a one, and the *Interrupt on Async Advance* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the *Interrupt on Async Advance* bit. |
| Reserved | 31:6 | | These bits are reserved and should be zero |

| FRINDEX –RW - 32 bits - [EOR_Reg : EHCI_EOR + 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Frame Index | 13:0 | 0h | When this bit is a one, and the Interrupt on *Async Advance* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt on *Async Advance* bit. |
| Reserved | 31:14 | | These bits are reserved and should be zero |

| CTRLDSSEGMENT –RW - 32 bits - [EOR_Reg : EHCI_EOR + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CTRLDSSEGMENT | 31:0 | 0h | This 32-bit register corresponds to the most significant address bits [63:32] for all EHCI data structures. If the *64-bit Addressing Capability* field in HCCPARAMS is a zero, then this register is not used. Software cannot write to it and a read from this register will return zeros.<br>If the *64-bit Addressing Capability* field in HCCPARAMS is a one, then this register is used with the link pointers to construct 64-bit addresses to EHCI control data structures. This register is concatenated with the link pointer from either the *PERIODICLISTBAS*E, *ASYNCLISTADD*R, or any control data structure link field to construct a 64-bit address.<br>This register allows the host software to locate all control data structures within the same 4 Gigabyte memory segment. |

| PERIODICLISTBASE –RW - 32 bits - [EOR_Reg : EHCI_EOR + 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 11:0 | | These bits are reserved. Must be written as 0s. During runtime, the values of these bits are undefined. |
| Base Address | 31:12 | 000h | These bits correspond to memory address signals [31:12], respectively. |

| ASYNCLISTADDR –RW - 32 bits - [EOR_Reg : EHCI_EOR + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 4:0 | | These bits are reserved and their value has no effect on operation. |
| Link Pointer Low (LPL) | 31:5 | 00h | These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). |

| CONFIGFLAG –RW - 32 bits - [EOR_Reg : EHCI_EOR + 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Configure Flag (CF) | 0 | 0b | Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below:<br>0b = Port routing control logic default-routes each port to an implementation dependent classic host controller.<br>1b = Port routing control logic default-routes all ports to this host controller. |
| Reserved | 31:1 | | These bits are reserved and should be set to zero. |

| PORTSC (1-N_PORTS) –RW - 32 bits - [EOR_Reg : EHCI_EOR + (44h~68h)] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Current Connect Status | 0 | 0b | 1 = Device is present on port.<br>0 = No device is present.<br>This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if *Port Power* is zero. [Read-only] |
| Connect Status Change | 1 | 0b | 1 = Change in Current Connect Status.<br>0 = No change.<br>Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). Software sets this bit to 0 by writing a 1 to it. This field is zero if *Port Power* is zero. |
| Port Enabled/Disabled | 2 | 0b | 1 = Enable.<br>0 = Disable<br>Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled (0b) downstream propagation of data is blocked on this port, except for reset. This field is zero if *Port Power* is zero. |
| Port Enable/Disable Change | 3 | 0b | 1 = Port enabled/disabled status has changed.<br>0 = No change.<br>For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2. Software clears this bit by writing a 1 to it. This field is zero if *Port Power* is zero. |
| Over-current Active | 4 | 0b | 1 = This port currently has an over-current condition.<br>0 = This port does not have an over-current condition.<br>This bit will automatically transition from a one to a zero when the over current condition is removed. |

| PORTSC (1-N_PORTS) –RW - 32 bits - [EOR_Reg : EHCI_EOR + (44h~68h)] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Force Port Resume | 6 | 0b | 1 = Resume detected/driven on port.<br>0 = No resume (K-state) detected/driven on port.<br>This functionality defined for manipulating this bit depends on the value of the *Suspend* bit. For example, if the port is not suspended *(Suspend* and *Enabled* bits are a one) and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the *Port Change Detect* bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. This field is zero if *Port Power* is zero. |
| Suspend | 7 | 0b | 1 = Port in suspend state.<br>0 = Port not in suspend state.<br>Port Enabled Bit and Suspend bit of this register define the port states as follows:<br>Bits [Port Enabled, Suspend] Port State<br>0X Disable<br>10 Enable<br>11 Suspend<br>When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when:<br>- Software sets the *Force Port Resume* bit to a zero (from a one).<br>- Software sets the *Port Reset* bit to a one (from a zero).<br>If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are undefined. This field is zero if *Port Power* is zero. |

| PORTSC (1-N_PORTS) –RW - 32 bits - [EOR_Reg : EHCI_EOR + (44h~68h)] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port Reset | 8 | 0b | 1 = Port is in Reset.<br>0 = Port is not in Reset.<br>When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Note: when software writes this bit to a one, it must also write a zero to the *Port Enable* bit.<br><br>Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the *Port Enable* bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2ms of software writing this bit to a zero.<br><br>The *HCHalted* bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the<br>*HCHalted* bit is a one. This field is zero if *Port Power* is zero. |
| Reserved | 9 | | This bit is reserved for future use, and should return a value of zero when read. |
| Line Status | 11:10 | | These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one.<br>The encoding of the bits are:<br>**Bits[11:10] USB State Interpretation**<br>00b SE0 Not Low-speed device, perform EHCI reset<br>10b J-state Not Low-speed device, perform EHCI reset<br>01b K-state Low-speed device, release ownership of port<br>11b Undefined Not Low-speed device, perform EHCI reset.<br>This value of this field is undefined if *Port Power* is zero.<br>[Read-only] |
| Port Power | 12 | | The function of this bit depends on the value of the *Port Power Control (PPC*) field in the HCSPARAMS register. The behavior is as follows:<br>**PPC PP Operation**<br>0b 1b RO - Host controller does not have port power control switches. Each port is hard-wired to power.<br>1b 1b/0b RW - Host controller has port power control switches.<br>This bit represents the current setting of the switch (0 = off, 1 = on). When power is not available on a port (i.e.<br>*PP* equals a 0), the port is non-functional and will not report attaches, detaches, etc.<br>When an over-current condition is detected on a powered port and *PPC* is a one, the *PP* bit in each affected port may be transitioned by the host controller from a 1 to 0 (removing power from the port).<br>[Read-write or Read-only] |

| PORTSC (1-N_PORTS) –RW - 32 bits - [EOR_Reg : EHCI_EOR + (44h~68h)] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Port Owner | 13 | 1b | This bit unconditionally goes to a 0b when the *Configured* bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the *Configured* bit is zero. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port. |
| Port Indicator Control | 15:14 | 00b | Writing to this bit has no effect if the *P_INDICATOR* bit in the HCSPARAMS register is a zero. If *P_INDICATOR* bit is a one, then the bit encodings are:<br>**Bit Value Meaning**<br>00b Port indicators are off<br>01b Amber<br>10b Green<br>11b Undefined<br>Refer to the USB Specification Revision 2.0 for a description on how these bits are to be used. This field is zero if *Port Power* is zero. |
| Port Test Control | 19:16 | 0000b | When this field is zero, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0110b - 1111b are reserved):<br>**Bits Test Mode**<br>0000b Test mode not enabled<br>0001b Test J_STATE<br>0010b Test K_STATE<br>0011b Test SE0_NAK<br>0100b Test Packet<br>0101b Test FORCE_ENABLE |
| Wake on Connect Enable | 20 | 0b | Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if *Port Power* is zero. |
| Wake on Disconnect Enable | 21 | 0b | Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if *Port Power* is zero. |
| Wake on Over-current Enable | 22 | 0b | Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if *Port Power* is zero. |
| Reserved | 31:23 | | Reserved |

| Packet Buffer Threshold Values – RW - 32 bits - [EOR_Reg : EHCI_EOR + 84h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IN Threshold | 7:0 | 10h | The PCI transaction starts when threshold of internal FIFO for receive packet is reached.<br>The value represents multiple of 8 bytes – 10h means 128 bytes. The smallest acceptable value is 08h (64 bytes). |
| Reserved | 15:8 | | Reserved |
| OUT Threshold | 23:16 | 60h | The transmit packet starts at UTMI interface when threshold of internal FIFO for transmit packet is reached.<br>The value represents multiple of 8 bytes – 10h means 128 bytes. The smallest acceptable value is 08h (64 bytes). |
| Reserved | 31:24 | | Reserved |

| USB PHY Status 0 – RW - 32 bits - [EOR_Reg: EHCI_EOR + 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PORT0_PHYStatus | 7:0 | 00h | Read only. PHY Status of Port0 |
| PORT1_PHYStatus | 15:8 | 00h | Read only. PHY Status of Port1 |
| PORT2_PHYStatus | 23:16 | 00h | Read only. PHY Status of Port2 |
| PORT3_PHYStatus | 31:24 | 00h | Read only. PHY Status of Port3 |
| Note: PORTx_PHYStatus[7:0] = { 0, RCKSEL, DUTYADJ[2:0], HSADJ[2:0] } where x=0 ~ 3 | | | |

| USB PHY Status 1 – RW - 32 bits - [EOR_Reg: EHCI_EOR + 8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PORT4_PHYStatus | 7:0 | 00h | Read only. PHY Status of Port4 |
| PORT5_PHYStatus | 15:8 | 00h | Read only. PHY Status of Port5 |
| PORT6_PHYStatus | 23:16 | 00h | Read only. PHY Status of Port6 |
| PORT7_PHYStatus | 31:24 | 00h | Read only. PHY Status of Port7 |

| USB PHY Status 2 – RW - 32 bits - [EOR_Reg: EHCI_EOR + 90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 15:0 | | Reserved |
| PORT8_PHYStatus | 23:16 | 00h | Read only. PHY Status of Port8 |
| PORT9_PHYStatus | 31:24 | 00h | Read only. PHY Status of Port9 |

| UTMI Control – RW - 32 bits - [EOR_Reg: EHCI_EOR + 94h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VControl | 6:0 | 0h | Control PHY setting<br><br>**Group-0** (VControlModeSel=0)<br>VControl[6:0] = {RCKSEL, DUTYADJ[2:0], HSADJ[2:0]}<br>- HSADJ : HS TX current adjustment<br>000 : -10%<br>001 : -5%<br>100 : 0%<br>101 : +5%<br>110 : +10%<br>- DUTYADJ: adjust clk480 (in analog PHY) duty cycle from range 40-60% to 60-40%.<br>- RCKSEL : to select the RCK fall into 50% or 57% of the eye<br>0 – 50% (center) of the eye<br>1 – 64% of the eye to increase setup time<br>**Group-1** (VControlModeSel =1)<br>VControl[6:0] = {Reserved, TESTMODE[3:0] } |
| VControlModeSel | 7 | 0b | To select PHY Vcontrol group0/1. |
| Reserved | 11:8 | | Reserved |
| VLoadB | 12 | 1b | Update PHY control mode (active load)<br>0: Load the new VControl value to PHY/common block<br>1: Only VControlModeSel value to PHY will be updated for selecting different PHY status group (see PHY status registers, EOR_Reg x88 ~ x90). But VControl[6:0] value inside PHY won't get affected. |
| Port Number | 16:13 | 0h | Select the corresponding port PHY or common block to load the VControl bits.<br>0000 – Port0<br>0001 – Port1<br>0010 – Port2<br>……<br>1001 – Port9<br>1010 ~ 1110 : Reserved , no effect<br>1111 – Common block |

| UTMI Control – RW - 32 bits - [EOR_Reg: EHCI_EOR + 94h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VBusy | 17 | 0b | RO – To block software write to [16:8] when port router is updating the field. |
| Reserved | 31:18 | | Reserved |

| BIST Control / Loopback Test – RW - 32 bits - [EOR_Reg : EHCI_EOR + 98h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | Reserved |
| Enable Loop Back test | 8 | 0b | Enable external USB Port Loop back test.<br>The Loop Back test is to set one port to TX mode (Test Packet mode) and one port in RX mode (Test SE0_NAK). Please reference to PORTSCx[19:16] control the port into TX or RX mode. |
| Loopback Test Status | 9 | 0b | Read Only.<br>Loop back status.<br>0: CRC Error on Loop Back Receiving Data<br>1: Good CRC on Loop Back Receiving data |
| Loopback Test Done | 10 | 0b | Read Only.<br>Indicate Loop back test done. |
| Reserved | 31:11 | 00000h | |

| EOR MISC Control – RW - 32 bits - [EOR_Reg : EHCI_EOR + 9Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 11:0 | 000h | Reserved |
| EHCI Power Saving Enable | 12 | 0b | Enable power saving clock gating. When enabled, dynamic clock gating is enabled when EHCI is not at operational mode. The clock goes to all memory module will be gated off, and the internal bus clock also gets gated off unless the connection interrupt is detected. |
| Reserved | 31:13 | 00000h | Reserved |

| USB Common PHY Calibration – RW - 32 bits - [EOR_Reg: EHCI_EOR + A0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ComCalBus | 6:0 | xx | Enables power saving clock gating (this was original at bit-31). When enabled, dynamic clock gating is enabled when EHCI is not at operational mode. The clock goes to all memory module will be gated off, The blink clock also is gated off unless the connection interrupt is detected. |
| Reserved | 7 | 0b | Reserved |
| NewCalBus | 15:8 | 00h | New calibration bus signed value. Bit-15 is the signed bit. |
| UsbCommonCalibration | 16 | 0b | If set, the PHY's calibration value in bit[6:0] is returned to the PHY ports.<br>If clear, the value after adjustment is returned to the PHY ports. |
| AddToCommonCalibration | 17 | 0b | If set, the signed NewCalBus is added to the ComCalBus and returned to the PHY ports. Any overflow is clamped to all ones. Any underflow is clamped to all zeros.<br>If clear, the NewCalBus (bit-14:8) replaces the ComCalBus and returns to the PHY ports. |
| Reserved | 31:18 | 0000h | Reserved |
| Note:<br>1. The equation for calibration resistor is as follows: Rcal = 1/ [1/59.4 + CalValue/(1.05*3.8k ohm)], where the CalValue is the final 7 bits of calibration setting send to PHY.<br>2. The total termination resistance value for HS USB D+/D- should include another 5 ohm resistance from FS driver. | | | |

#### *2.2.3.4 USB2.0 Debug Port Registers*

This block of registers is memory-mapped. The base offset, Dbase, is directly defined in DBUG_PRT

Control register (EHCI_PCI_CFG xE4[28:16], default = 0E0h), regardless of the value in register (MEM_Reg: 00h).

| Registers Name | Offset Address |
|---|---|
| Control / Status | DBase + 00h |
| USB PIDs | DBase + 04h |
| Data Buffer | DBase + (08h~0Ch) |
| Device Address | DBase + 10h |

| Control / Status – RW - 32 bits - [DBUG_Reg : DBase + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Data Length | 3:0 | 0h | For write operations, this field is set by software to indicate to the hardware how many bytes of data in *Data Buffer* are to be transferred to the console when *Write/Read#* is set when software sets *Go*. A value of 0h indicates that a zero-length packet should be sent. A value of 1-8 indicates 1-8 bytes are to be transferred. Values 9-Fh are illegal and how hardware behaves if used is undefined.<br>For read operations, this field is set by hardware to indicate to software how many bytes in *Data Buffer* are valid in response to software setting *Go* when *Write/Read#* is cleared. A value of 0h indicates that a zero length packet was returned. (The state of *Data Buffer* is not defined.) A value of 1-8 indicates 1-8 bytes were received. Hardware is not allowed to return values 9-Fh. The transferring of data always starts with byte 0 in the data area and moves toward byte 7 until the transfer size is reached. |
| Write/Read# | 4 | 0b | Software sets this bit to indicate that the current request is a write and clears it to indicate a read. |
| Go | 5 | 0b | Software sets this bit to cause the hardware to perform a request. Writing this bit to a 1 when the bit is already set may result in undefined behavior. Writing a 0 to this bit has no effect. When set, the hardware clears this bit when the hardware sets the *Done* bit. (Completion of a request is indicated by the *Done* bit.) |
| Error/Good# | 6 | 0b | Read Only<br>Updated by hardware at the same time it sets the *Done* bit. When set it indicates that an error occurred. Details of the error are provided in the *Exception* field. When cleared, it indicates that the request terminated successfully. |
| Exception | 9:7 | 000b | Read Only<br>This field indicates the exception when *Error/Good#* is set. This field cannot be cleared by software. Reset default = 000b.<br>**Value** **Meaning**<br>000b None<br>001b Transaction error: indicates the USB2 transaction had an error (CRC, bad PID, timeout, etc.)<br>010b HW error. Request was attempted (or in progress) when the port was suspended or reset.<br>011b-111b Reserved |
| In Use | 10 | 0b | Set by software to indicate that the port is in use. Cleared by software to indicate that the port is free and may be used by other software. (This bit has no affect on hardware.) |
| Reserved | 15:11 | | Reserved |
| Done | 16 | 0b | RWC<br>This bit is set by HW to indicate that the request is complete. Writing a 1 to this bit will clear it. Writing a 0 to this bit has no effect. |
| Reserved | 27:17 | | Reserved |

| Control / Status – RW - 32 bits - [DBUG_Reg : DBase + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Enabled | 28 | 0b | This bit is a one if the debug port is enabled for operation. Software can clear this by writing a zero to it. The controller clears the bit for the same conditions where hardware clears the Port Enable/Disable Change bit (in the PORTSC register). (Note: this bit is not cleared when System Software clears the *Port Enabled/Disabled* bit (in the PORTSC register). Software can directly set this bit, if the port is already enabled in the associated Port Status and Control register (this is HW enforced). |
| Reserved | 29 | | Reserved |
| Owner | 30 | 0b | When debug software writes a one to this bit, the ownership of the debug port is forced to the EHCI controller (i.e. Immediately taken away from the companion controller). If the port was already owned by the EHCI controller, then setting this bit is has no effect. This bit overrides all of the ownership related bits in the standard EHCI registers. Reset default = 0. Note that the value in this bit may not affect the value reported in the *Port Owner* bit in the associated PORTSC register. |
| Reserved | 31 | | Reserved |

| USB PIDs – RW - 32 bits - [DBUG_Reg : DBase + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Token PID | 7:0 | 00h | The debug port controller sends this PID as the Token PID for each USB transaction. Software will typically set this field to either IN, OUT or SETUP PID values. Reset default = undefined. |
| Send PID | 15:8 | 00h | The debug port controller sends this PID to begin the data packet when sending data to USB (i.e. *Write/Read#* is asserted). Software will typically set this field to either DATA0 or DATA1 PID values. Reset default = undefined. |
| Received PID | 23:16 | 00h | Read Only<br>The debug port controller updates this field with the received PID for transactions in either direction. When the controller is sending data (*Write/Read#* is asserted), this field is updated with the handshake PID that is received from the device. When the host controller is receiving data (*Write/Read#* is not asserted), this field is updated with the data packet PID (if the device sent data), or the handshake PID (if the device NAKs the request). This field is valid when the controller sets the *Done* bit. Reset default = undefined. |
| Reserved | 31:24 | | Reserved |

| Data Buffer – RW - 64 bits - [DBug_Reg : DBase + 08h/0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Data Buffer | 63:0 | 00000000_00000000h | The least significant byte is accessed at offset 08h and the most significant byte is accessed at offset 0Fh. Each byte in *Data Buffer* can be individually accessed. *Data Buffer* must be written with data before software initiates a write request. For a read request, *Data Buffer* contains valid data when *Done* is set, *Error/Good#* is cleared, and *Data Length* specifies the number of bytes that are valid. Reset default = undefined. |

| Device Address – RW - 32 bits - [DBUG_Reg : DBase + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| USB Endpoint | 3:0 | 1h | 4-bit field that identifies the endpoint used by the controller for all Token PID generation. |
| Reserved | 7:4 | | Reserved |
| USB Address | 14:8 | 7Fh | 7-bit field that identifies the USB device address used by the controller for all Token PID generation. |
| Reserved | 31:15 | | Reserved |

## 2.3 SMBus Module and ACPI Block (Device 20, Function 0)

Some registers in the SMBus/ACPI PCI configuration space (PCI_reg, see *section 2.3.1*) contain controls and settings for a number of blocks within the SB600. *Figure 4* below shows these blocks, with their affected functions and the associated PCI_reg registers.

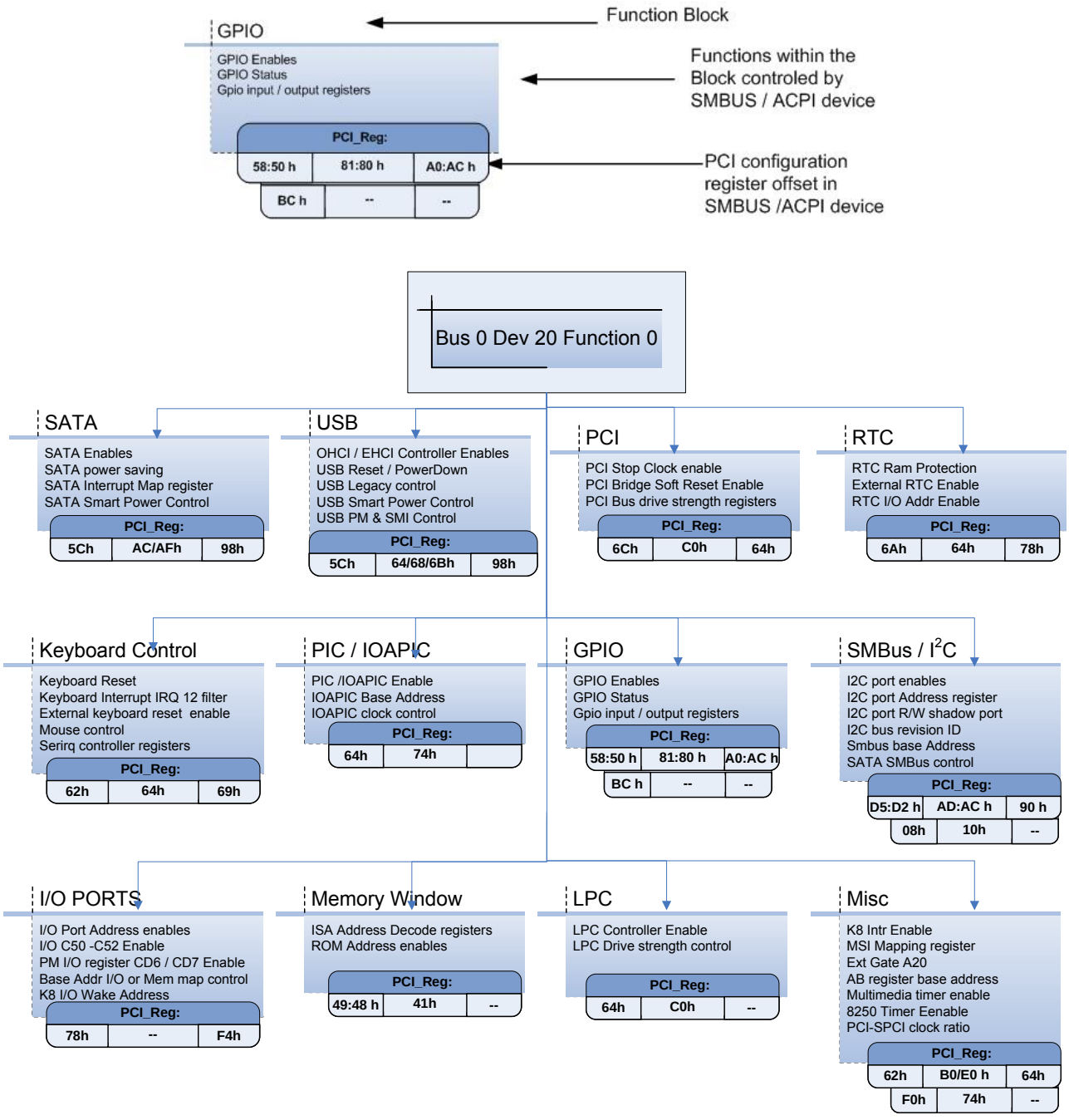



**Figure 4 SMBus/ACPI PCI Configuration Space Function Block Association**

### *2.3.1 PCI Configuration Registers and Extended Registers*

#### *2.3.1.1 PCIE Configuration Registers*

| Register Name | Configuration Offset |
|---|---|
| VendorID | 00h |
| DeviceID | 02h |
| Command | 04h |
| STATUS | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| BIST | 0Fh |
| Base Address 0 | 10h |
| Base Address 1 | 14h |
| Base Address 2 | 18h |
| Base Address 3 | 1Ch |
| Base Address 4 | 20h |
| Base Address 5 | 24h |
| Cardbus CIS Pointer | 28h |
| Subsystem Vendor | 2Ch |
| Subsystem ID | 2Eh |
| Expansion ROM Base Address | 30h |
| Capability Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Min_Gnt | 3Eh |
| Max_Lat | 3Fh |
| PCI Control | 40h |
| MiscFunction | 41h |
| DmaLimit | 42h |
| DmaEnhanceEnable | 43h |
| ISA Address Decode Control Register #1 | 48h |
| ISA Address Decode Control Register #2 | 49h |
| GPIO_52_to_49_Cntrl | 50h |
| GPIO_56_to_53_Cntrl | 52h |
| GPIO_60_to_57_Cntrl | 54h |
| GPIO_64_to_61_Cntrl | 56h |
| GPIO_73_to_70_Cntrl | 5Ah |
| SmartPowerControl1 | 5Ch |
| SmartPowerControl2 | 5Dh |
| MiscEnable | 62h |
| AzIntMap | 63h |
| Features Enable | 64h |
| SerialIrqControl | 69h |
| RTCProtect | 6Ah |
| USB Reset | 6Bh |
| TestMode | 6Ch |
| IoApic_Conf | 74h |
| IoAddrEnable | 78h |
| GPIO_69_68_66_65_Cntrl | 7Eh |
| GPIO_3_to_0_Cntrl | 80h |
| GPIO_32_31_14_13_Cntrl | 82h |
| Smbus Base Address | 90h |
| IDE_GPIO_Cntrl | A0h |

| Register Name | Configuration Offset |
|---|---|
| IDE_GPIO_In | A4h |
| GPIO_48_47_46_37_Cntrl | A6h |
| GPIO_12_to_4_Cntrl | A8h |
| SATA_Cntrl | ACh |
| SataIntMap | AFh |
| MSI_Mapping_Capability | B0h |
| PciIntGpio | BCh |
| UsbIntMap | BEh |
| IoDrvSth | C0h |
| I2CbusConfig | D2h |
| I2CCommand | D3h |
| I2CShadow1 | D4h |
| I2Cshadow2 | D5h |
| I2CBusRevision | D6h |
| MSI_Weight | E0h |
| AB_REG_BAR | F0h |
| WakeIoAddr | F4h |
| MwaitID | F6h |
| MwaitSts | F7h |
| ExtendedAddrPort | F8h |
| ExtendedDataPort | FCh |

| VendorID - R - 16 bits - [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VendorID | 15:0 | 1002h | Vendor ID |
| Vendor ID register: Vendor Identification | | | |

| DeviceID - R - 16 bits - [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DeviceID | 31:16 | 4385h | Device ID |
| Device ID register: Device Identification Number | | | |

| Command- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I/O Space | 0 | 1b | This bit controls a device's response to IO space accesses. A value of 1 enables it and a value of 0 disables it. Since this module does claim certain legacy IO cycles, this bit is default to 1. |
| Memory Space | 1 | 1b | This bit controls a device's response to memory space accesses. A value of 1 enables it and a value of 0 disables it. Since this module does claim certain memory cycles if BIOS is strapped to the PCI bus, this bit is default to 1. |
| Bus Master | 2 | 0b | A value of 0 disables the device from generating PCI accesses. A value of 1 allows it to behave as a bus master. ACPI/SMBus does not have PCI master and so it is always 0. [Read-only] |
| Special Cycle | 3 | 0b | A value of 0 causes the devices to ignore all special cycle operations. A value of 1 allows the device to monitor Special Cycle operations. This module does not respond to special cycle and so this is hardcoded to 0 |
| Memory Write & Invalidate Enable | 4 | 0b | This bit is an enable bit for using the Memory Write and Invalidate command. This module will not generate this command and so it is always 0. [Read-only] |
| VGA Palette Snoop | 5 | 0b | This bit controls how VGA compatible and graphics devices handle accesses to VGA pallette registers. This does not apply to this module and so it is always 0. [Read-only] |

| Command- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Parity Error Response | 6 | 0b | This bit controls the device's response to parity errors. When the bit is set, the device must take its normal action when a parity error is detected. When the bit is 0, the device must ignore any parity errors that it detects and continue normal operation. |
| Wait Cycle Control | 7 | 0b | This bit is used to control whether or not a device does address/data stepping. This module does not use address stepping. [Read-only] |
| SERR# Enable | 8 | 0b | This bit is an enable bit for SERR# driver. A value of 0 disables the SERR# and a value of 1 enables it. |
| Fast Back-to-Back Enable | 9 | 0b | This bit indicates whether device is fast back-to-back capable. ACPI/SMbus does not support this function and so this bit is always 0. [Read-only] |
| Reserved | 15:10 | 00h | |
| PCI Command register | | | |

| STATUS- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | | |
| MSI Mapping Capability | 4 | 1/0b | [Read-only] This bit indicates whether the device can support MSI mapping. For K8 system this device is MSI mapping capable so default value is 1; for P4 system this device does not support MSI mapping so default value is 0. |
| 66 MHz Capable | 5 | 1b | This bit indicates whether the device can support 66 MHz. This device is 66 MHz capable. [Read-only] |
| UDF Supported | 6 | 0b | This bit indicates whether the device supports user definable feature. This module does not support this feature and so it is always 0. [Read-only] |
| Fast Back-to-Back Capable | 7 | 0b | This bit indicates whether the device is capable of fast back-to-back cycles. This module does not support this feature and so it is always 0. [Read-only] |
| Data Parity Error Detected | 8 | 0b | Set to 1 if the Parity Error Response bit is set, and the module has detected PERR# asserted while acting as a PCI master (regardless PERR# was driven by this module). |
| DEVSEL Timing | 10:9 | 01b | These bits encode the timing of DEVSEL#. This module will always respond in medium timing and so these bits are always 11. |
| Signaled Target Abort | 11 | 0b | This bit is set by a slave device whenever it terminates a cycle with a Target-Abort. |
| Received Target Abort | 12 | 0b | This bit is set by a master device whenever its transaction is terminated with a Target-Abort. |
| Received Master Abort | 13 | 0b | This bit is set by a slave device whenever it terminates its transaction with Master-Abort. |
| Signaled System Error | 14 | 0b | This bit is set by device whenever the device asserts SERR#. |
| Detected Parity Error | 15 | 0b | This bit is set by device whenever it detects a parity error, even if parity error handling is disabled. |
| PCI device status register | | | |

| Revision ID/Class Code- R - 32 bits - [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RevisionID | 7:0 | 11h /<br>12h /<br>13h | This field reflects the ASIC revision.<br>11h : For ASIC revision A11<br>12h : For ASIC revision A12<br>13h : For ASIC revision A13<br>For ASIC revisions after A13, by default this field will read 13h still. However, if SMBUS PCI config 70h bit 8 is set to 1, a hidden revision ID can be read from this field. |
| Class Code | 31:8 | 0C0500h | 0C0500h denotes a SMBUS controller. |
| Revision ID/Class Code register | | | |

| Cache Line Size- R - 8 bits - [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | This register specifies the system cacheline size. This module does not use Memory Write and Invalidate command and so this register is not applicable. It is hardcoded to 0. |
| Cache line size register | | | |

| Latency Timer- R - 8 bits - [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer | 7:0 | 00h | This register specifies the value of the Latency Timer. This is not used in this module and so it is always 0. |
| Latency timer register | | | |

| Header Type- R - 8 bits - [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 80h | This device is a multifunction device. |
| Header type register | | | |

| BIST- R - 8 bits - [PCI_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST | 7:0 | 00h | The module has no built-in self-test and so this is always 0. |
| BIST register | | | |

| Base Address 0- R - 32 bits - [PCI_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO/Memory | 0 | 1b | 1 = IO<br>0 = Memory |
| Reserved | 3:1 | 000b | |
| SmBusBaseAd | 31:4 | 0000000h | SMBus Base Address |
| Base Address 0 register | | | |

| Base Address 1- R - 32 bits - [PCI_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 9:0 | 000h | Hardwired to 0; memory map only |
| MultiMediaTimerBaseAddr | 31:10 | 000000h | High Precision Event Timer (also called Multi-media Timer) base address. |
| Base Address 1 register | | | |

| Base Address 2- R - 32 bits - [PCI_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 2 | 31:0 | 0000_0000h | Not used and is hardcoded to 0. |

| Base Address 2- R - 32 bits - [PCI_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 2 register | | | |

| Base Address 3- R - 32 bits - [PCI_Reg: 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 3 | 31:0 | 0000_0000h | Not used and is hardcoded to 0. |
| Base Address 3 register | | | |

| Base Address 4- R - 32 bits - [PCI_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 4 | 31:0 | 0000_0000h | Not used and is hardcoded to 0. |
| Base Address 4 register | | | |

| Base Address 5- R - 32 bits - [PCI_Reg: 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 5 | 31:0 | 0000_0000h | Not used and is hardcoded to 0. |
| Base Address 5 register | | | |

| Cardbus CIS Pointer- R - 32 bits - [PCI_Reg: 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cardbus CIS Pointer | 31:0 | 0000_0000h | Not used and is hardcoded to 0. |
| Cardbus CIS Pointer register | | | |

| Subsystem Vendor ID- W - 16 bits - [PCI_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Write once. |
| Subsystem Vendor ID register | | | |

| Subsystem ID- W - 16 bits - [PCI_Reg: 2Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem ID | 15:0 | 0000h | Write once. |
| Subsystem ID register | | | |

| Expansion ROM Base Address - R - 8 bits - [PCI_Reg: 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Expansion ROM Base Address | 7:0 | 00h | Not used and is hardcoded to 0. |
| Expansion ROM Base Address register | | | |

| Capability Pointer - R - 8 bits - [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability Pointer | 7:0 | B0/00h | For K8 system default value is B0h; for P4 system default value is 00h. |
| Capability Pointer register | | | |

| Interrupt Line - R - 8 bits - [PCI_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | This module does not generate interrupt. This register is hardcoded to 0. |
| Interrupt Line register | | | |

| Interrupt Pin – R - 8 bits - [PCI_Reg: 3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 00h | This register specifies which interrupt pin the device issues. This module does not generate interrupt but contains the actual interrupt controller. This register is hardcoded to 0. |
| Interrupt Pin register | | | |

| Min_Gnt - R - 8 bits - [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Min_Gnt | 7:0 | 00h | This register specifies the desired settings for Latency Timer values. Value of 0 indicates that the device has no major requirements for the setting. This value is hardcoded to 0. |
| Min_Gnt register | | | |

| Max_Lat - R - 8 bits - [PCI_Reg: 3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Max_Lat | 7:0 | 00h | This register specifies the desired settings for Latency Timer values. Value of 0 indicates that the device has no major requirements for the setting. This value is hardcoded to 0. |
| Max_Lat register | | | |

| PCI Control- RW - 8 bits - [PCI_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 00b | |
| KB2RstEnable | 2 | 0b | When set, KeyBoard reset (KBRST#) pin will generate a system wide reset (ARST#) for P4 system; for K8 system, additional control by PMIO 66h Bit 5 determines whether INIT# or ARST# is generated |
| Reserved | 7:3 | 0h | |
| PCI Control register | | | |

| MiscFunction- RW - 8 bits - [PCI_Reg: 41h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| ExtraROM AddrEnable2 | 1 | 0b | This bit only has meaning if xbus ROM is used. If this bit is set, addresses between FFF80000h to FFFDFFFFh will be directed to the ROM interface |
| Reserved | 2 | 0b | |
| WatchDogDecodeEn | 3 | 0b | Enables watchdog decode |
| ExtraROM AddrEnable1 | 4 | 0b | This bit is meaningful if ROM interface is strapped to the xbus ROM (sits on PCI bus). If this bit is set, addresses between 0E0000h to 0EFFFFh will be directed to the ROM interface. |
| MiscfuncEnable | 5 | 0b | When set, this module will decode cycles to IO C50, C51, C52: GPM controls. |
| Reserved | 7:6 | 00b | |
| MiscFunction register | | | |

| DmaLimit- RW - 8 bits - [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DmaBurstLimit | 6:0 | 00h | Enables the amount of burst data the legacy DMA engine can sustain before it should give up the internal bus. |
| DmaLimitEnable | 7 | 0h | This is another enhancement to the legacy DMA engine. In the original design, certain DMA request (such as Infrared) will cause the legacy DMA engine to dominate the internal bus for a very long time (up to 512 bytes) and thereby causing long latency for other devices. Setting this bit will cause the legacy DMA engine to limit its transfer per burst based on bits [6:0] |
| DmaLimit register | | | |

| DmaPrefetchEnable RW - 8 bits - [PCI_Reg: 43h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DmaPrefetchEnable | 0 | 1b | Legacy read DMA prefetch function enable.<br>1 – Enable<br>0 – Disable<br>When set, the DMA engine will keep the data inside the FIFO, even though the requesting device has deasserted the DMA request. When the device requests data again, the DMA engine will have data available instead of having to fetch data from the memory again. Note this enhancement only applies to channel 0, 1, 2, and 3. It has no effect on channel 5, 6, or 7 |
| Reserved | 7:1 | 00h | |
| DmaPrefetchEnable register | | | |

| ISA Address Decode Control Register #1- RW - 8 bits - [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Window 0 | 0 | 1b | 896K to 960K PCI enable |
| Window 1 | 1 | 1b | 640K to 768K PCI enable |
| Window 2 | 2 | 1b | 512K to 640K PCI enable |
| Window 3 | 3 | 1b | 0K to 512K PCI enable |
| Reserved | 7:4 | 0h | |
| ISA Address Decode Control Register #1: This register defines the enable bits for four memory segments. If the enable bit is set to 1, an ISA master or DMA access to the memory segment associated with that bit is forwarded to the internal bus. The SB600 does not have any ISA master, because the bus is internal; however, it may affect DMA transfers with the LPC module. Software should set all these bits to 1's. | | | |

| ISA Address Decode Control Register #2- RW - 8 bits - [PCI_Reg: 49h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Window 0 | 0 | 1b | C0000h to C3FFFh |
| Window 1 | 1 | 1b | C4000h to C7FFFh |
| Window 2 | 2 | 1b | C8000h to CBFFFh |
| Window 3 | 3 | 1b | CC000h to CFFFFh |
| Window 4 | 4 | 1b | D0000h to D3FFFh |
| Window 5 | 5 | 1b | D4000h to D7FFFh |
| Window 6 | 6 | 1b | D8000h to DBFFFh |
| Window 7 | 7 | 1b | DC000h to DFFFFh |
| ISA Address Decode Control Register #1: This register defines the PCI enable bits for four memory segments. If the enable bit is set to 1, an ISA master or DMA access to the memory segment associated with that bit is forward to the internal bus. The SB600 does not have any ISA master, because the bus is internal; however, it may affect the DMA transfers with the LPC module. Software should set all these bits to 1's. | | | |

| GPIO_52_to_49_Cntrl - RW – 16 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port; providing the corresponding enable bits (7:4) are set to 0<br>Bit[0] for GPIO49/FANOUT2<br>Bit[1] for GPIO50/FANIN0<br>Bit[2] for GPIO51/FANIN1<br>Bit[3] for GPIO52/FANIN2 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |
| GPIO_52_to_49_Cntrl register | | | |

| GPIO_56_to_53_Cntrl - RW – 16 bits - [PCI_Reg: 52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port; providing the corresponding enable bits (7:4) are set to 0<br>Bit[0] for GPIO53/VIN0<br>Bit[1] for GPIO54/VIN1<br>Bit[2] for GPIO55/VIN2<br>Bit[3] for GPIO56/VIN3 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |
| GPIO_56_to_53_Cntrl register | | | |

| GPIO_60_to_57_Cntrl - RW – 16 bits - [PCI_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding enable bits (7:4) are set to 0<br>Bit[0] for GPIO57/VIN4<br>Bit[1] for GPIO58/VIN5<br>Bit[2] for GPIO59/VIN6<br>Bit[3] for GPIO60/VIN7 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |
| GPIO_60_to_57_Cntrl register | | | |

| GPIO_64_to_61_Cntrl - RW – 16 bits - [PCI_Reg: 56h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding enable bits (7:4) are set to 0<br>Bit[0] for GPIO61/TEMPIN0<br>Bit[1] for GPIO62/TEMPIN1<br>Bit[2] for GPIO63/TEMPIN2<br>Bit[3] for GPIO64/TEMPIN3 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |

| GPIO_64_to_61_Cntrl - RW – 16 bits - [PCI_Reg: 56h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_64_to_61_Cntrl register | | | |

| ASFSMbusIoBase- RW - 16 bits - [PCI_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ASFSMBusEnable | 0 | 0h | 0 – Disable ASF controller<br>1 – Enable ASF controller |
| Reserved | 3:1 | 000b | |
| ASFSMBase | 15:4 | FFFh | ASF SM bus controller Io base address |

| GPIO_73_to_70_Cntrl - RW – 16 bits - [PCI_Reg: 5Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port ; providing the corresponding GPIO_OUT_En# and GPIO_Enable are set appropriately<br>Bit[0] for GPIO70/REQ3#<br>Bit[1] for GPIO71/REQ4#<br>Bit[2] for GPIO72/GNT3#<br>Bit[3] for GPIO73/GNT4# |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| GPIO_Enable | 15:12 | 0h | GPIO function enable for each of the GPIO port<br>When set, the pin becomes GPIO<br>0: GPIO disabled<br>1: GPIO enabled |
| GPIO_73_to_70_Cntrl register | | | |

| SmartPowerControl1A - RW – 8 bits - [PCI_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CheckLpc | 0 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if LPC is idle |
| CheckAz | 1 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if HD audio is idle |
| CheckAc97 | 2 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if AC97 is idle |
| CheckPciBridge | 3 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if HD Audio PCIBridge is idle |
| CheckUsb | 4 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if USBis idle |
| CheckSata | 5 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if SATA is idle |
| CheckIde | 6 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if IDE is idle |
| CheckC3 | 7 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if CPU is in C3 state |
| SmartPowerControl1A register | | | |

| SmartPowerControl1B - RW – 8 bits - [PCI_Reg: 5Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CheckVIN0 | 0 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN0 has reached or passed the threshold |
| CheckVIN1 | 1 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN1 has reached or passed the threshold |
| CheckVIN2 | 2 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN2 has reached or passed the threshold |
| CheckVIN3 | 3 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN3 has reached or passed the threshold |
| CheckVIN4 | 4 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN4 has reached or passed the threshold |
| CheckVIN5 | 5 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN5 has reached or passed the threshold |
| CheckVIN6 | 6 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN6 has reached or passed the threshold |
| CheckVIN7 | 7 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt if VIN7 has reached or passed the threshold |
| SmartPowerControl1B register | | | |

| MiscEnable- RW - 8 bits - [PCI_Reg: 62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IRQ1_Filter | 0 | 0b | Keyboard interrupt filter enable (IRQ1)<br>Filtering is done so that the first rising edge of IRQ1 would cause the IRQ1 going to the PIC (8259 programmable interrupt controller) to go asserted, but subsequent changes to IRQ1 would not have any effect on the IRQ1 going to the PIC until a access to the keyboard was done (I/O read of port 60.<br>The Main effect is that software could mask IRQ1, do several accesses / commands to the keyboard controller that subsequently cause numerous IRQ1's, do one final I/O read access of port 60 and know that when IRQ1 was unmasked that no pending keyboard interrupt would be generated. |
| IRQ12_Filter | 1 | 0b | Mouse Interrupt Filter Enable (IRQ12)<br>0 – Disable IRQ12 filtering<br>1 – Enable IRQ12 filtering |
| K8_INTR | 2 | 0b | K8 INTR Enable (BIOS should set this bit after PIC initialization)<br>0 – Disable K8 INTR message<br>1 – Enable K8 INTR message |
| MT3_Set | 3 | 0b | If this bit is set, K8 INTR NMI Message Type field (Bit3) is forced to be 1; otherwise K8 INTR NMI Message Type is controlled by MT3_Auto. Recommended method is to use MT3_Auto bit.. In AMD K8 system, all interrupts are sent to CPU via messages. In MP base (such as Linux), the message may need to be in certain format. |
| MT3_Auto | 4 | 0b | If this bit is set, K8 INTR NMI Message Type field (Bit3) is 1 if APIC is also active; otherwise K8 INTR NMI Message Type is 0 |

| MiscEnable- RW - 8 bits - [PCI_Reg: 62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| USB_Fast_SMI_Disable | 5 | 0b | For K8 system, legacy USB can request SMI# to be sent out early before IO completion. Some applications may have problem with this feature. BIOS should set this bit to 1 to disable the feature. |
| Reserved | 6 | 0b | |
| IDE_GPIO_Enable | 7 | 0b | If this bit is set to 1, the IDE bus is configured as GPIO |
| MiscEnable register | | | |

| AzIntMap- RW - 8 bits - [PCI_Reg: 63h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AzIntMap | 2:0 | 110b | Interrupt routing table for HD Audio. Setting this register routes the HD audio's interrupt to the specific PCI interrupt before it is routed to the interrupt controller<br>000 – INTA#<br>001 – INTB#<br>010 – INTC#<br>011 – INTD#<br>100 – INTE#<br>101 – INTF#<br>110 – INTG#<br>111 – INTH# |
| Reserved | 7:3 | 00000b | |

| Features Enable- RW - 32 bits - [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PIC_Enable | 0 | 1b | PIC (8259) Programmable Interrupt Controller enable<br>0 - I/O cycles to master PIC:20,21, slave PIC:a0, a1, ELCR registers 4D0, 4D1h, and the PCI interrupt Mapping Registers (C00, C01), and Numberic Coprocessor Error Register (IRQ13) (0F0h) are not accepted.<br>1 - (default) I/O cycles to APIC are not decoded but I/O cycles to these above addresses will be positive decoded on PCI and run to the internal 8259 PIC |
| Timer_Enable | 1 | 1b | 0 – I/O cycles to timers/counter (040-043h) will not be claimed on ISA<br>1 – I/O cycles to timers/counter will be claimed on ISA and run to the internal 8254 Timer/Counter |
| PMIO_Register Enable | 2 | 1b | Power management enable register<br>0 – I/O cycles to Power management registers (CD6 and CD7h) will not be claimed<br>1 – I/O cycles to Power management registers will be claimed and run to the internal Power Management logic (BIOS should always set it to 1) |
| Ioapic_enable | 3 | 0b | When set, this block will decode ioapic address |
| CheckOwnReq | 4 | 0b | If set, the SB600 will check its own REQ# as the PCI_ACTIVE signal in addition to BMREQ# |
| BmReqEn | 5 | 0b | BMREQ# enable |
| Reserved | 6 | 0b | |
| XIOAPIC_ENA | 7 | 1b | XIOAPIC enable; this bit is only valid if bit 3 is set. |
| GEVENT5_ENA | 8 | 0b | BIOS should always set this bit to 1 to enable GEVENT5. |
| Ext_KBRST_EnB | 9 | 0b | Enable external KB_RST# input. When set to 0, GEVENT[1] is used as KBRST# input |
| MultiMediaTimerIrqEn | 10 | 0b | High Precision Event Timer (also called Multimedia Timer) interrupt enable |
| Ext_A20En | 11 | 0b | Enable external Ga20In input. When set to 1, GEVENT[0] is used as Ga20In input |

| Features Enable- RW - 32 bits - [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Smi_Gevent_En | 12 | 0b | Enable all the events with the capability of doing both SMI# and SCI to SMI# assertion. If enabled, an assertion at any of the event inputs will cause SMI# to be asserted if SCI EN is not set.<br>0 – Disable<br>1 – Enable |
| Intr_block_En | 13 | 0b | Applicable in the PIC system. When enabled, it will block any pending interrupt for approximately 500ns after IntrAck cycle from the host.<br>0 – Disable<br>1 – Enable |
| ApicPort02Swap | 14 | 0b | When set, port 0 of APIC is connected to output of the PIC and port 2 is connected to IRQ0 (timer0)<br>When clear, port 0 of APIC is connected to IRQ0 (timer0) and port 2 is connected to the output of the PIC. Software should set this bit to conform with MP spec. |
| UsbSmiEn | 15 | 0b | USB SMI# enable:<br>1 = Enable<br>0 = Disable |
| Serr2Smi_En | 16 | 0b | Enable SERR# to SMI# assertion<br>0 – Disable<br>1 – Enable |
| Gevent1_en0 | 17 | 0b | GEVENT group 0 enable (GPM[7] to SMI#/SCI enable) to ACPI function |
| Gevent1_en1 | 18 | 0b | GEVENT group 1 enable (GPM[6] to SMI#/SCI enable) to ACPI function |
| Gevent1_en2 | 19 | 0b | GEVENT group 2 enable (GPM[5:4] to SMI#/SCI enable) to ACPI function |
| LpcEnable | 20 | 0b | 1 – Enable lpc controller<br>0 – Disable lpc controller |
| RtcSelect | 21 | 0b | Writing this bit with 1 will toggle the selection between Internal RTC and External RTC which is set via strap bit.<br>Reading this bit returns the internal/external mode:<br>1 = External RTC<br>0 = Internal RTC |
| Gevent1_en3 | 22 | 0b | GEVENT group 3 enable (ExtEvent[1:0], PCIePme) to ACPI function |
| Gevent1_en4 | 23 | 0b | GEVENT group 4 enable (GPM[3:0]) to ACPI function |
| Reserved | 24 | | |
| DmaVerifyEn | 25 | 0b | With LPC is replacing the ISA bus, software needs to set this bit in order to for DMA verify to work properly |
| IRQ1MergeEn | 26 | 0b | This is the old method to merge the normal IRQ1 with the USB legacy IRQ1 function. Since this is no longer needed, software should always leave this bit with 0. |
| IRQ12MergeEn | 27 | 0b | This is the old method to merge the normal IRQ12 with the USB legacy IRQ12 function. Since this is no longer needed, software should always leave this bit with 0. |
| SB_ClkStpEn | 29:28 | 00b | SB_ClkStpEn[0] enables PciStpB to stop primary PCI clock.<br>SB_ClkStpEn[1] enables PciStpB to stop secondary PCI clock |
| UsbA20En | 30 | 0b | USB A20 enable |
| UsbLegacyIrqEn | 31 | 0b | Enable for IRQ1/12 from USB (BIOS should always set it to 1) |
| Features Enable register | | | |

| UsbEnable - RW - 8 bits - [PCI_Reg: 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| EHCI_enable | 0 | 1b | Set to 1 to enable EHCI |
| OHCI_0_enable | 1 | 1b | Set to 1 to enable OHCI_0 |
| OHCI_1_enable | 2 | 1b | Set to 1 to enable OHCI_1 |

| UsbEnable - RW - 8 bits - [PCI_Reg: 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| OHCI_2_enable | 3 | 1b | Set to 1 to enable OHCI_2 |
| OHCI_3_enable | 4 | 1b | Set to 1 to enable OHCI_3 |
| OHCI_4_enable | 5 | 1b | Set to 1 to enable OHCI_4 |
| Reserved | 7:6 | 00b | |
| UsbEnable register | | | |

| SerialIrqControl- RW - 8 bits - [PCI_Reg: 69h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NumStartBits | 1:0 | 00b | This field defines the number of clocks in the start frame.<br>Start Frame Width = 4 + 2 * NumStartBits |
| NumSerIrqBits | 5:2 | 0h | Total number of serial IRQ's = 17 + NumSerIrqbits<br>0 - 17 serial IRQ's (15 IRQ, SMI#, + IOCHK#)<br>1 - 18 serial IRQ's (15 IRQ, SMI#, IOCHK#, INTA#)<br>...<br>15 - 32 serial IRQ's<br>The SB600 serial IRQ can support 15 IRQ#, SMI#, IOCHK#, INTA#, INTB#, INTC#, and INTD#.<br>When serial SMI# is used, BIOS will need to check SIO (or device that generates serial SMI#) for status. |
| SerIrqMode | 6 | 0b | 0 - Continuous mode<br>1 - Active (quiet) mode |
| SerialIrqEnable | 7 | 0b | Setting this bit to 1 enable the serial IRQ function |
| SerialIrqControl register | | | |

| RTCProtect- RW - 8 bits - [PCI_Reg: 6Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RTCProtect38_3F | 0 | 0b | When set, RTC RAM index 38:3Fh will be locked from read/write. This bit can only be written once. |
| RTCProtectF0_FF | 1 | 0b | When set, RTC RAM index F0:FFh will be locked from read/write. This bit can only be written once. |
| RTCProtectE0_EF | 2 | 0b | When set, RTC RAM index E0:EFh will be locked from read/write. This bit can only be written once. |
| RTCProtectD0_DF | 3 | 0b | When set, RTC RAM index D0:DFh will be locked from read/write. This bit can only be written once. |
| RTCProtectC0_CF | 4 | 0b | When set, RTC RAM index C0:CFh will be locked from read/write. This bit can only be written once. |
| Reserved | 7:5 | 000b | |
| RTCProtect register | | | |

| USB Reset- RW - 8 bits - [PCI_Reg: 6Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Force Reset to USB Host Controllers | 4:0 | 00h | These are software control bits that force the reset of the USB host controllers. |
| Force USB Port PHY Power Down | 5 | 0b | Forces USB PHY into power down mode. |
| Force PHY PLL Power Down | 6 | 0b | Forces USB PHY PLL into power down mode. |
| Force USB Port PHY Reset | 7 | 0b | Forces USB PHY reset. |
| USB Reset register | | | |

| TestMode- RW - 16 bits - [PCI_Reg: 6C] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA_Timing | 0 | 0b | To be used by BIOS only; when set, legacy DMA will insert 1 extra idle clock in between requests. Software should always set this bit. |
| TestMode | 4:1 | 0h | These bits are for testing only. Software should not write to these bits. |
| PCIB_SReset_En Mask | 5 | 0b | When set, PCIB_SReset_En (x3e bit 22 of PCI Bridge) will be writable. |
| TestMode | 15:6 | 000h | These bits are for testing only. Software should not write to these bits. |
| TestMode register | | | |

| IoApic_Conf- RW - 32 bits - [PCI_Reg: 74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 000b | |
| Mem_IO_Map | 3 | 1b | Base address mapping<br>1 = memory map<br>0 = IO map |
| Reserved | 4 | 0b | |
| IoApic_Addr | 31:5 | 1111_<br>1110_<br>1100_<br>0000_<br>0000_<br>0000_<br>000b | Base address for IOAPIC |
| IoApic_Conf register. | | | |

| IoAddrEnable - RW - 32 bits - [PCI_Reg: 78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DmaAddr_En | 0 | 1b | 0x000:0x01F, 0x080:0x08F, 0x0C0:0xCF, 0x0D0:0x0DF, 0x40B, 0x4D6, |
| PitAddr_En | 1 | 1b | 0x40,0x41, 0x42, 0x43 |
| NmiAddr_En | 2 | 1b | 0x70 |
| RtcAddr_En | 3 | 1b | 0x71 |
| Misc_Enable1 | 4 | 1b | 0xC14 |
| Misc_Enable2 | 5 | 1b | 0xC49, 0xC4A |
| Misc_Enable3 | 6 | 1b | 0xC52 |
| Misc_Enable4 | 7 | 1b | 0xC6C |
| Misc_Enable5 | 8 | 1b | 0xC6F |
| PM_Addr_Enable | 9 | 1b | 0xCD6,0 xCD7 |
| Reserved | 10 | 0b | |
| Cms_Enable | 11 | 1b | Address 0xC50, 0xC51 |
| Reserved | 13:12 | 00b | |
| Port92Enable | 14 | 1b | Port 92 enable |
| Reserved | 31:15 | 00000h | |
| IoAddrEnable Register: When a bit is set, this block will decode the corresponding address. If the bit is cleared, this block will not claim the corresponding address. This is to allow the legacy port to be behind the PCI bridge. | | | |

| GPIO_69_68_66_65_Cntrl - RW – 16 bits - [PCI_Reg: 7Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding bits [7:4] are enabled<br>Bit[0] for GPIO65/BMREQ#<br>Bit[1] for GPIO66/LLB#<br>Bit[2] for GPIO68/LDRQ1#<br>Bit[3] for GPIO69/RTC_IRQ# |

| GPIO_69_68_66_65_Cntrl - RW – 16 bits - [PCI_Reg: 7Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |
| GPIO_69_68_66_65_Cntrl register | | | |

| GPIO_3_to_0_Cntrl - RW – 16 bits - [PCI_Reg: 80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding bits [7:4] are enabled<br>Bit[0] for GPIO0/ SSMUXSEL<br>Bit[1] for GPIO1/ROM_CS#<br>Bit[2] for GPIO2/SPKR<br>Bit[3] for GPIO3/FANOUT0 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate<br>For GPIO1, this is applicable only if we are not using the external PCI bus as the ROM interface. |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| Reserved | 15:12 | 0h | |
| GPIO_3_to_0_Cntrl register | | | |

| GPIO_32_31_14_13_Cntrl - RW – 16 bits - [PCI_Reg: 82h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding bits [7:4] and [15:12] are enabled<br>Bit[0] for GPIO13/LAN_RST#<br>Bit[1] for GPIO14/ROM_RST#<br>Bit[2] for GPIO31/SPI_HOLD#<br>Bit[3] for GPIO32/SPI_CS# |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| GPIO_Enable | 15:12 | 0h | GPIO function enable for each of the GPIO port<br>0: GPIO disabled<br>1: GPIO enabled |
| GPIO_32_31_14_13_Cntrl register | | | |

| Smbus Base Address - R – 32 bits - [PCI_Reg: 90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO/Memory | 0 | 1b | 1 = IO<br>0 = Memory |
| Reserved | 3:1 | 000b | |
| SmBusBaseAd | 31:4 | 0000000h | SMBus Base Address |
| Smbus Base Address register (also accessible through 10h) | | | |

| SmartPowerControl2A - RW – 8 bits - [PCI_Reg: 98h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CheckLpc | 0 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if LPC is idle |
| CheckAz | 1 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if azalia (HD audio) is idle |
| CheckAc97 | 2 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if AC97 is idle |
| CheckPciBridge | 3 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if azalia PCIBridge is idle |
| CheckUsb | 4 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if USBis idle |
| CheckSata | 5 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if SATA is idle |
| CheckIde | 6 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if IDE is idle |
| CheckC3 | 7 | 0b | If SmartVoltEnable2 is set and this bit is also set, the SmartPower2 function will only assert SmartVolt2 if CPU is in C3 state |
| SmartPowerControl2A register | | | |

| SmartPowerControl2B - RW – 8 bits - [PCI_Reg: 99h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CheckVIN0 | 0 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN0 has reached or passed the threshold |
| CheckVIN1 | 1 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN1 has reached or passed the threshold |
| CheckVIN2 | 2 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN2 has reached or passed the threshold |
| CheckVIN3 | 3 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN3 has reached or passed the threshold |
| CheckVIN4 | 4 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN4 has reached or passed the threshold |
| CheckVIN5 | 5 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN5 has reached or passed the threshold |
| CheckVIN6 | 6 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN6 has reached or passed the threshold |
| CheckVIN7 | 7 | 0b | If SmartVoltEnable is set and this bit is also set, the SmartPower function will only assert SmartVolt2 if VIN7 has reached or passed the threshold |
| SmartPowerControl2B register | | | |

| SmartPowerControl2C - RW – 8 bits - [PCI_Reg: 9Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmartVoltIdleTime2 | 6:0 | 0h | Amount of “idle” time (in 2us increment) the SmartPower2 function should wait before it should assert SmartVolt 2 |

| SmartPowerControl2C - RW – 8 bits - [PCI_Reg: 9Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmartVoltEnable2 | 7 | 0b | Enable bit for the SmartPower2 function. When set, the logic will monitor the logic (defined by 98h). If all of the corresponding modules are idle, |
| SmartPowerControl2C register<br>SmartVolt function is meant to provide a mechanism to control the external power supply in order to reduce additional system power consumption. For example, software can set SmartPowerControl2A[6] and SmartPowerControl2A[7]. Whenever CPU enters C3 state and IDE (PATA) controller is not active, this function will assert GPIO5. System design can use this signal to control the power supply to reduce the ATA power by 5~10%. Another example is to connect an ambient light sensor to one of the VIN inputs. When the circuit has detected the ambient light is below certain threshold, this function can automatically dim the LCD back light. | | | |

| IDE_GPIO_Cntrl – RW - 32 bits - [PCI_Reg:A0] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 15:0 | 0000h | When the IDE bus is used as GPIO, these bits control the output of each IDE data bit; providing the corresponding bits [31:16] are enabled |
| GPIO_Out_En# | 31:16 | FFFFh | When the IDE bus is used as GPIO, these bits control the output enable of each IDE data bit.<br>0 = Enable<br>1 = Tristate |
| IDE_GPIO_Cntrl register | | | |

| IDE_GPIO_In – R - 16 bits - [PCI_Reg: A4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Status | 15:0 | ---- | When the IDE bus is used as GPIIO, these are the read ports for each IDE data bit. |
| IDE_GPIO_In register | | | |

| GPIO_48_47_46_37_Cntrl - RW – 16 bits - [PCI_Reg: A6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 3:0 | 0h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding bits [7:4] and [15:12] are enabled<br>Bit[0] for GPIO37/DPSLP_OD#<br>Bit[1] for GPIO46/AZ_SDIN3<br>Bit[2] for GPIO47/SPI_CLK<br>Bit[3] for GPIO48/FANOUT1 |
| GPIO_Out_En# | 7:4 | Fh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GPIO_Status | 11:8 | - | GPIO input status for each of the GPIO port |
| GPIO_Enable | 15:12 | 0h | GPIO function enable for each of the GPIO port<br>0: GPIO disabled<br>1: GPIO enabled<br>Bit[13] [15] no effect<br>Use PM_Reg: 60h bit [2] to configure GPIO48/FANOUT1. |
| GPIO_48_47_46_37_Cntrl register | | | |

| GPIO_12_to_4_Cntrl – RW – 32 bits - [PCI_Reg: A8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO_Out | 7:0 | 00h | Write 1 to set and 0 to clear each of the GPIO port providing the corresponding bits [15:8], [25], [31:30] are enabled<br>Output for GPIO[12:11][9:4]<br>Bit[0] for GPIO4/SMARTVOLT<br>Bit[1] for GPIO5/SHUTDOWN/SMARTVOLT2<br>Bit[2] for GPIO6/GNI#<br>Bit[3] for GPIO7/VGATE<br>Bit[4] for GPIO8/DDC1_SDA<br>Bit[5] for GPIO9/DDC1_SCL<br>Bit[6] for GPIO11/SPI_DO<br>Bit[7] for GPIO12/SPI_DI |
| GPIO_Out_En# | 15:8 | FFh | GPIO output port enable for each of the GPIO port<br>0: Output = GPIO_Out<br>1: Output = tristate |
| GpioIn | 23:16 | - | GPIO input status for each of the GPIO port |
| GPIO10_Out | 24 | 0b | Write 1 to set and 0 to clear GPIO10/SATA_IS0# |
| GPIO10_Out_En# | 25 | 1b | GPIO output port enable for GPIO10/SATA_IS0#<br>0: Output = GPIO10_Out<br>1: Output = tristate |
| GPIO10_Status | 26 | - | GPIO input status for GPIO10/SATA_IS0# |
| Reserved | 29:27 | 000b | |
| GPIO_Enable | 31:30 | 0h | GPIO function enable for GPIO[12:11]<br>0: GPIO disabled<br>1: GPIO enabled |
| GPIO_12_to_4_Cntrl register | | | |

| SATA_Cntrl - RW – 16 bits - [PCI_Reg: ACh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO67_Out | 0 | 0b | Write 1 to set and 0 to clear GPIO67/SATA_ACT# |
| GPIO67_Out_En# | 1 | 1b | GPIO output port enable for GPIO67/SATA_ACT#<br>0: Output = GPIO67_Out<br>1: Output = tristate |
| GPIO67_Status | 2 | - | GPIO input status for GPIO67/SATA_ACT# |
| GPIO67_Enable | 3 | 0b | GPIO function enable for GPIO67/SATA_ACT#<br>0: GPIO disabled<br>1: GPIO enabled |
| SMI_CMD_action | 4 | 1b | If this bit is enabled, SMI_CMD or SLP_trap will cause SMI being sent to host regardless of EOS status. Otherwise, SMI is sent only when EOS=1. |
| Reserved | 7:5 | 0h | |
| SataEnable | 8 | 1b | SATA enable |
| SataSmbusCfg | 10:9 | 00b | SATA SMBus configuration.<br>00: Reserved<br>01: Disable SATA SMBus<br>10: Reserved<br>11: Enable SATA SMBu<br>There is only one SATA SMBus (I2C) interface for the two SATA controllers. The SATA I2C interface is only used for characterization purposes. Use either an external I2C master or the on-chip SMBus as I2C master to talk to the SATA I2C target. |
| Reserved | 12:11 | 00b | |
| SataPsvEn | 13 | 1b | SATA power saving enable |
| Reserved | 14 | 0b | |
| HiddenMsiEnable | 15 | 0b | Setting this bit will make PCI_Reg:B0h, bit 16 to show up as 1. |

| SATA_Cntrl - RW – 16 bits - [PCI_Reg: ACh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtendIntrToWakeTime | 18:16 | 000b | This is used in K8 system to extend the interrupt break event status. Whenever there is an apic interrupt, this logic will extend the break event status by the amount of time defined by this register. This is to avoid potential race condition between CPU issuing the C1e command and the SB seeing an interrupt. If CPU tries to enter C state before the extension time expires, SB will break out from the C state. Each count represents 2 microsecond increment and it has an uncertainty of 2 microseconds. |
| DrqMaskEn | 19 | 0b | Setting this bit to 1 will cause the legacy DMA request to be blocked if the DMA channel has not been initialized properly. This bit is applicable to ASIC revision A21 and above. |
| IdeIrqFilterEn | 20 | 0b | Setting this bit to 1 will cause narrow pulses (less than several A-link clocks wide) on the IDE IRQ line to be filtered out. This means that no interrupt will be generated. This bit is applicable to ASIC revision A21 and above. |
| TmrIrqEnhanceDisable | 21 | 0b | This bit should be set to 1 for the normal operation of the 8254 timer. This setting is required by ASIC revisions A11, A12, and A13. |
| PIC_APIC_coexist | 22 | 0b | This bit should be set to 1 if PIC and APIC are to be enabled at the same time in K8 system. There is no harm done even if they don't coexist at the same time. |
| Reserved | 25:23 | 000b | |
| SataIntMap | 28:26 | 000b | SATA interrupt mapping to PCI interrupt<br>000 - INTA#, 001 - INTB#, 010 - INTC#, 011 - INTD#, 100 - INTE#, 101 - INTF#, 110 - INTG#, 111 - INTH# |
| Reserved | 31:29 | 000b | |
| SATA_Cntrl register | | | |

| MSI Mapping Capability - R - 32 bits - [PCI_Reg: B0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 08h | This is a HyperTransport capability list item. |
| Capability Pointer | 15:8 | 00h | This is the end of capability list. |
| MsiEnable | 16 | 0b | MSI enable programmable through PCI_Reg: ADh bit 7 |
| Fixed | 17 | 1b | The address for mapping MSIs is fixed at 0000_0000_FEEx_xxxxh. |
| Reserved | 26:18 | 000h | |
| Capability Type | 31:27 | 10101b | This is an MSI Mapping Capability block. |
| MSI Mapping Capability register | | | |

| PciIntGpio - RW - 16 bits - [PCI_Reg: BCh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PciIntGpioOut | 3:0 | 0h | Output data for each PCI INT# GPIO providing bits [7:4] and [15:12] are enabled |
| PciIntGpioEnB | 7:4 | Fh | Output enable for each PCI INT# GPIO (active low) |
| PciIntGpioStatus | 11:8 | - | Input status for each PCI INT# GPIO [Read Only] |
| PciIntIsGpio | 15:12 | 0h | Set to 1 to use PCI interrupt INTH/G/F/E# as GPIO |
| PciIntGpio register | | | |

| UsbIntMap - RW - 16 bits - [PCI_Reg: BEh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| UsbInt1Map | 2:0 | 000b | (OHCI0) UsbInt1 interrupt mapping to PCI interrupt |
| UsbInt2Map | 5:3 | 001b | (OHCI1 & OHCI3) UsbInt2 interrupt mapping to PCI interrupt |
| Reserved | 7:6 | 00b | |
| UsbInt3Map | 10:8 | 010b | (OHCI2 & OHCI4) UsbInt3 interrupt mapping to PCI interrupt |
| UsbInt4Map | 13:11 | 011b | (EHCI) UsbInt4 interrupt mapping to PCI interrupt |

| UsbIntMap - RW - 16 bits - [PCI_Reg: BEh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 15:14 | 00b | |
| UsbIntMap register<br>Encoding:<br>000 - INTA#, 001 - INTB#, 010 - INTC#, 011 - INTD#, 100 - INTE#, 101 - INTF#, 110 - INTG#, 111 - INTH# | | | |

| IoDrvSth - RW - 32 bits - [PCI_Reg: C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IoDrvSth_AD | 1:0 | 11b | IO drive strength (bits [2:1]); together with bit 26 make up the drive strength control for AD[31:0], CBE0#, CBE1#, CBE2#, CBE3# and PAR pads. |
| IoDrvSth_Cntrl | 3:2 | 11b | IO drive strength (bits [2:1]); together with bit 26 make up the drive strength control for A_RST#, FRAME#, IRDY#, DEVSEL#, TRDY#, LOCK#, STOP#, PERR#, SERR#, CLKRUN# and PCIRST# pads |
| IoDrvSth_GNT | 5:4 | 11b | IO drive strength (bits [2:1];; together with bit 27 make up the drive strength control for GNT#[2:0], GNT3#, GNT4#, GNT5#, and GNT6# pads |
| IoDrvSth_ClkGrpA | 7:6 | 11b | IO drive strength (bits [2:1];; together with bit 27 make up the drive strength control for PCICLK[6:5] pads |
| IoDrvSth_ClkGrpB | 9:8 | 11b | IO drive strength (bits [2:1];; together with bit 27 make up the drive strength control for PCICLK[4:1] pads |
| IoDrvSth_Lpc | 12:11 | 11b | IO drive strength (bits [2:1];; together with bit 28 make up the drive strength control for LAD[3:0], LFRAME#, and LDRQ#[1:0] pads |
| IoDrvSth_Ac97 | 13:12 | 11b | IO drive strength for AC_SYNC, AC_SDOUT and SPDIF_OUT pads |
| IoDrvSth_Int | 15:14 | 11b | IO drive strength for INTA#, INTB#, INTC#, INTD#, INTE#, INTF#, INTG# and INTH# pads |
| IoDrvSth_Req | 17:16 | 11b | IO drive strength for REQ#[1:0], REQ3#, REQ4#, REQ5# and REQ6# pads when these pads are configured as GPIO |
| IoDrvSth_GpioA | 19:18 | 11b | IO drive strength for BMREQ#, EXTEVENT1#, GPIO[0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 13, 37], GPOC[0:1], DPRSLPVR, |
| IoDrvSth_GpioB | 21:20 | | IO drive strength for GPIO[3, 48:52], |
| IoDrvSth_Misc | 23:22 | 11b | IO drive strength (bits [2:1];; together with bit 29 make up the drive strength control for CPU_STP#/DPSLP_3V#, GA20IN, KBRST#, SERIRQ and SATA_ACT# pads |
| IoDrvSth_Ide | 25:24 | 11b | IO drive strength for IDE pads |
| IoDrvSth_AD_0 | 26 | 1b | Bit 0 of IoDrvSth_AD |
| IoDrvSth_GNT_0 | 27 | 1b | Bit 0 of IoDrvSth_GNT |
| IoDrvSth_Lpc_0 | 28 | 1b | Bit 0 of IoDrvSth_Lpc |
| IoDrvSth_Misc_0 | 29 | 1b | Bit 0 of IoDrvSth_Misc |
| Reserved | 31:30 | 11b | |

IoDrvSth: Each three bit field controls the number of P and N transistors enabled in the final stage of the output driver for the designated pads. By controlling the number of transistors enabled, the designer can optimize the drive characteristics of signals based on the topology of their specific design. For drive strength with 3 bit control, the drive strength table is shown below. The values are non-linear and values for each bit field of this register are as follows:

| Value | Relative Strength | Description |
|---|---|---|
| 111b | 100% | All P and N transistors are enabled when the pad drives |
| 110b | 88% | Approximately 88% of the P and N transistors are enabled |
| 101b | 77% | Approximately 77% of the P and N transistors are enabled |
| 100b | 66% | Approximately 66% of the P and N transistors are enabled |
| 011b | 66% | Approximately 66% of the P and N transistors are enabled |
| 010b | 55% | Approximately 55% of the P and N transistors are enabled |
| 001b | 44% | Approximately 44% of the P and N transistors are enabled |
| 000b | 33% | Approximately 33% of the P and N transistors are enabled |

| IoDrvSth - RW - 32 bits - [PCI_Reg: C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| For control with only two bits, one can assume the four drive strength settings (corresponding to 25, 50, 75, and 100% respectively). | | | |

For the PCI interface, bits [3:0] and 26 control the AD and PCI control signals, The table below contains recommendations based on empirical data collected from a live system:

| S2 | S1 | S0 | Percent Drive Strength | Loads | SMBUS PCI Register, Index C0h |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 33% | For 0-1 loads | Bits[1:0] = 01; Bits[3:2] = 00, bit 26 = 0 |
| 0 | 0 | 1 | 44% | For 2 loads | Bits[1:0] = 10; Bits[3:2] = 00, bit 26 = 1 |
| 0 | 1 | 0 | 55% | For 3-4 loads | Bits[1:0] = 01; Bits[3:2] = 01, bit 26 = 0 |
| 0 | 1 | 1 | 66% | For 4-5 loads | Bits[1:0] = 01; Bits[3:2] = 01, bit 26 = 1 |

For LPC bus, the drive strength is determined by SMBUS PCI C0h, bits [11:10] and [28]

| S2 | S1 | S0 | Percent Drive Strength | Loads | SMBUS PCI Register, Index C0h |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 33% | For 0-1 loads | Bits[11:10] = 00, bit 28 = 0 |
| 0 | 0 | 1 | 44% | For 2 loads | Bits[11:10] = 00, bit 28 = 1 |
| 0 | 1 | 0 | 55% | For 3-4 loads | Bits[11:10] = 01, bit 28 = 0 |
| 0 | 1 | 1 | 66% | For 4-5 loads | Bits[11:10] = 01, bit 28 = 1 |

| I2CbusConfig - RW - 8 bits - [PCI_Reg: D2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I2CController Enable | 0 | 0b | I2C controller host interface enable |
| I2CbusInterrupt | 1 | 0b | 0 : SMI#<br>1 : IRQ |
| Reserved | 7:2 | 00h | |
| I2CbusConfig register: Registers D2-D5 control the interface when this chip is the I2C slave. | | | |

| I2CCommand - RW - 8 bits - [PCI_Reg: D3h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I2Ccommand | 7:0 | 00h | I2C Host Slave Command; this value specifies the command value to be matched for I2C master accesses to the I2Ccontroller host slave interface. |
| I2CCommand register | | | |

| I2CShadow1- RW - 8 bits - [PCI_Reg: D4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Read/Write ShadowPort1 | 0 | 0b | Read/Write for Shadow Port 1<br>This bit must be programmed to 0 because I2C slave controller only responds to Word Write Transaction. |
| I2CslaveAddr1 | 7:1 | 00h | SMBus Slave Address for shadow port 1<br>This value specifies the address used to match against incoming I2C addresses for Shadow port 1. |

| I2CShadow1- RW - 8 bits - [PCI_Reg: D4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I2CShadow1 register | | | |

| I2Cshadow2- RW - 8 bits - [PCI_Reg: D5h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Read/Write ShadowPort2 | 0 | 0b | Read/Write for Shadow Port 2<br>This bit must be programmed to 0 because I2C slave controller only responds to Word Write Transaction. |
| I2CslaveAddr2 | 7:1 | 00h | SMBus Slave Address for shadow port 2<br>This value specifies the address used to match against incoming I2C addresses for Shadow port 2. |
| I2Cshadow2 register | | | |

| I2CBusRevision - RW - 8 bits - [PCI_Reg: D6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I2CbusRevision | 7:0 | 00h | Revision ID |
| I2CBusRevision register | | | |

| MSI_Weight - RW – 8 bits - [PCI_Reg: E0h] | | | |
|---|---|---|---|
| | **Bits** | **Default** | **Description** |
| MSI_weight | 5:0 | 100000b | Used by bios to tune MSI messaging priority |
| Reserved | 7:6 | 00b | |
| MSI_Weight register | | | |

| AB_REG_BAR - RW - 32 bits - [PCI_Reg: F0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AB_REG_BAR | 31:0 | 0000_0000h | Base Address for A-link Bridge Register |
| AB_REG_BAR register | | | |

| WakeIoAddr- RW - 16 bits - [PCI_Reg: F4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WakeIoAddr | 15:0 | 0000h | IO Address for C-State Wake-up by CPU (K8 only). The BIOS can program an address inside K8 and this location. The K8 can then use it to generate an IO write to tell SB to wake from C state (location inside K8 is TBD). |
| WakeIoAddr register | | | |

| MwaitID- RW - 8 bits - [PCI_Reg: F6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mwait_physical_ID | 3:0 | 0100b | This is used for P4 dual core system. Two physical CPU IDs with default values 00 and 01 to match with addr[19:18] of MWAIT and ADS_after_MWAIT. Usage TBD |
| Mwait_logical_ID | 7:4 | 0100b | This is used for P4 dual core system. Two logical CPU IDs with default values 00 and 01 to match with addr[17:16] of MWAIT and ADS_after_MWAIT. Usage TBD |
| MwaitID register | | | |

| MwaitSts- R - 8 bits - [PCI_Reg: F7h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mwait_cpu0_sts | 0 | 0b | Set to 1 by MWAIT with addr[19:18] = Mwait_physical_ID[1:0] and addr[17:16] = Mwait_logical_ID[1:0]. Cleared by ADS_after_MWAIT with the same addr[19:16]. |

| MwaitSts- R - 8 bits - [PCI_Reg: F7h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mwait_cpu1_sts | 1 | 0b | Set to 1 by MWAIT with addr[19:18] = Mwait_physical_ID[3:2] and addr[17:16] = Mwait_logical_ID[1:0]. Cleared by ADS_after_MWAIT with the same addr[19:16]. |
| Mwait_cpu2_sts | 2 | 0b | Set to 1 by MWAIT with addr[19:18] = Mwait_physical_ID[1:0] and addr[17:16] = Mwait_logical_ID[3:2]. Cleared by ADS_after_MWAIT with the same addr[19:16]. |
| Mwait_cpu3_sts | 3 | 0b | Set to 1 by MWAIT with addr[19:18] = Mwait_physical_ID[3:2] and addr[17:16] = Mwait_logical_ID[3:2]. Cleared by ADS_after_MWAIT with the same addr[19:16]. |
| Reserved | 7:4 | 0h | |
| MwaitSts register. This register is to be used for P4 dual core system. Usage TBD. | | | |

| ExtendedAddrPort- RW - 32 bits - [PCI_Reg: F8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtendedAddrPort | 31:0 | 00h | Address port for the extended register block |
| ExtendedAddrPort register.. | | | |

| ExtendedDataPort- RW - 32 bits - [PCI_Reg: FCh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtendedDataPort | 31:0 | 00h | Data port for the extended register block |
| ExtendedDataPort register. | | | |

#### *2.3.1.2 Extended Registers*

| Register Name | Configuration Offset |
|---|---|
| AudioPortConfig | 00h |
| AudioGpioControl | 04h |

| AudioPortConfig- RW - 32 bits - [Extend_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AzPort0Config | 1:0 | 01b | Port configuration for HD Audio and AC97 input port:<br>00 or 11 = GPIO port<br>01 = Set as ac97 port<br>10 = Set as HD Audio port |
| AzPort1Config | 3:2 | 01b | Port configuration for HD Audio and AC97 input port:<br>00 or 11 = GPIO port<br>01 = Set as ac97 port<br>10 = Set as HD Audio port |
| AzPort2Config | 5:4 | 10b | Port configuration for HD Audio and AC97input port:<br>00 or 11 = GPIO port<br>01 = Set as ac97 port<br>10 = Set as HD Audio port |
| AzPort3Config | 7:6 | 10b | Port configuration for HD Audio and AC97input port:<br>00 or 11 = GPIO port<br>01 = Set as ac97 port<br>10 = Set as HD Audio port |
| Reserved | 15:8 | | |
| AudioGpioIn0Status | 16 | - | When ACZ_SDIN0 is configured as GPIO, this bit returns the GPIO input status. |
| AudioGpioIn1Status | 17 | - | When ACZ_SDIN 1 is configured as GPIO, this bit returns the GPIO input status. |
| AudioGpioIn2Status | 18 | - | When ACZ_SDIN 2 is configured as GPIO, this bit returns the GPIO input status. |
| AudioGpioIn3Status | 19 | - | When ACZ_SDIN 3 is configured as GPIO, this bit returns the GPIO input status. |
| AzRstGpioIN | 20 | - | When AZ_RST# is configured as GPIO, this bit returns the GPIO input status. |
| Reserved | 23:21 | | |
| Ac97RstGpioIn | 24 | - | When AC_RST# is configured as GPIO, this bit returns the GPIO input status. |
| Ac97BclkGpioIn | 25 | - | When AC_BitClk is configured as GPIO, this bit returns the GPIO input status. |
| Ac97SyncGpioIn | 26 | - | When AC_Sync is configured as GPIO, this bit returns the GPIO input status. |
| Ac97DOutGpioIn | 27 | - | When AC_DataOut is configured as GPIO, this bit returns the GPIO input status. |
| SpdifGpioIn | 28 | - | When AC_SPDIF is configured as GPIO, this bit returns the GPIO input status. |
| Reserved | 31:29 | | |
| AudioPortConfig register. | | | |

| AudioGpioControl – RW - 32 bits - [Extend_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AudioGpioOut0OeB | 0 | 1b | When ACZ_SDIN0 is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| AudioGpioOut1OeB | 1 | 1b | When ACZ_SDIN1 is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| AudioGpioOut2OeB | 2 | 1b | When ACZ_SDIN2 is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| AudioGpioOut3OeB | 3 | 1b | When ACZ_SDIN3 is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| AzRstGpioOutOeB | 4 | 1b | When AZ_RST# is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| Reserved | 7:5 | | |
| Ac97RstGpioOutOeB | 8 | 1b | When AC_RST# is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| Ac97BclkGpioOeB | 9 | 1b | When AC_BitClk is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| Ac97SyncGpioOeB | 10 | 1b | When AC_Sync is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| Ac97DOutGpioOeB | 11 | 1b | When AC_ SDOut is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| SpdifGpioOeB | 12 | 1b | When AC_SPDIF is configured as GPIO, this bit represents the output enable.<br>1 – Tristate<br>0 – Enable |
| Reserved | 15:13 | | |
| AudioGpioOut0 | 16 | 0b | When ACZ_SDIN 0 is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| AudioGpioOut1 | 17 | 0b | When ACZ_SDIN 1 is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| AudioGpioOut2 | 18 | 0b | When ACZ_SDIN 2 is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| AudioGpioOut3 | 19 | 0b | When ACZ_SDIN 3 is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |

| AudioGpioControl – RW - 32 bits - [Extend_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AzRstGpioOut | 20 | 0b | When AZ_RST# is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| Reserved | 23:21 | | |
| Ac97RstGpioOut | 24 | 0b | When AC_RST# is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| Ac97BclkGpioOut | 25 | 0b | When AC_BITCLK is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| Ac97SyncGpioOut | 26 | 0b | When AC_SYNC is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| Ac97DOutGpioOut | 27 | 0b | When AC_SDOUT is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| SpdifGpioOut | 28 | 0b | When SPDIF_OUT is configured as GPIO, this bit represents the output value if the output is enabled.<br>1 – High<br>0 – Low |
| Reserved | 31:29 | | |
| AudioGpioControl register. | | | |

### *2.3.2 SMBus Registers*

| Register Name | Offset Address |
|---|---|
| SMBusStatus | 00h |
| SMBusSlaveStatus | 01h |
| SMBusControl | 02h |
| SMBusHostCmd | 03h |
| SMBusAddress | 04h |
| SMBusData0 | 05h |
| SMBusData1 | 06h |
| SMBusBlockData | 07h |
| SMBusSlaveControl | 08h |
| SMBusShadowCmd | 09h |
| SMBusSlaveEvent | 0A-0Bh |
| SlaveData | 0C-0Dh |
| SMBusTiming | 0Eh |

Note: The SMBus registers are located at the IO memory space base address defined by PCI configuration register 90-93h

| SMBusStatus - RW - 8 bits - [SMBUS:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| HostBusy | 0 | 0b | This bit indicates the SMBus controller is in the process of completing a command. When this bit is set, software should not access any other SMBus registers [Read-only] |
| SMBusInterrupt | 1 | 0b | This bit is set by hardware to indicate the completion of the last host command. This bit can be cleared by writing an 1 to it. |
| DeviceErr | 2 | 0b | This bit is set by hardware to indicate an error of one of the following: 1) illegal command field, 2) unclaimed cycle, 3) host device time-out. This bit can be cleared by writing an 1 to it. |
| BusCollision | 3 | 0b | This bit is set by hardware to indicate SMBus transaction collision; this bit can be cleared by writing an 1 to it. |
| Failed | 4 | 0b | This bit is set by hardware to indicate a failed bus transaction, set when SMBusControl.Kill bit is set. This bit is cleared by writing an 1 to it |
| Reserved | 7:5 | 000b | |

| SMBusSlaveStatus - RW - 8 bits - [SMBUS:01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SlaveBusy | 0 | 0b | This bit indicates the SMBus controller slave interface is in the process of receiving data. Software should not try to access any other SMBus register when this bit is set. [Read-only] |
| SlaveInit | 1 | 0b | Writing a 1 to this bit will initialize the slave. It is unnecessary to write it back to 0. A read from it will always return a 0. |
| SlaveStatus | 2 | 0b | This bit is set by hardware to indicate a slave cycle event match of the SMBus slave command and SMBus Slave Event match. This bit can be cleared by writing an 1 to it. |
| Shadow1Status | 3 | 0b | This bit is set by hardware to indicate a slave cycle address match of the SMB_Shadow1 port. This bit can be cleared by writing a 1 to it. |
| Shadow2Status | 4 | 0b | This bit is set by hardware to indicate a slave cycle address match of the SMB_Shadow2 port. This bit can be cleared by writing a 1 to it. |
| AlertStatus | 5 | 0b | This bit is set by hardware to indicate SMBALERT_ signal. This function is not supported. [Read-only] |

| SMBusSlaveStatus - RW - 8 bits - [SMBUS:01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:6 | 00b | |

| SMBusControl - RW - 8 bits - [SMBUS:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| InterruptEnable | 0 | 0b | Enable the generation of interrupts on the completion of current host transaction. |
| Kill | 1 | 0b | Stop the current host transaction in process |
| SMBusProtocol | 4:2 | 000b | 000 – Quick Read or Write<br>001 – Byte Read or Write<br>010 – Byte Data Read or Write<br>011 – Word Data Read or Write<br>100 – Reserved<br>101 – Block Read or Write<br>110 – Reserved<br>111 – Reserved |
| Reserved | 5 | 0b | |
| Start | 6 | 0b | Writing an 1 in this field initiates SMBus controller host interface to execute the command programmed in the SMBusProtocol field |
| Reserved | 7 | 0b | |

| SMBusHostCmd - RW – 8 bits - [SMBUS:03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusHostCmd | 7:0 | 00h | This field contains the data transmitted in the command field of SMBus host transaction |

| SMBusAddress - RW - 8 bits - [SMBUS:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusRdWr | 0 | 0b | 1 – Execute a Read command<br>0 – Execute a Write command |
| SMBusAddr | 7:1 | 00h | This field contains the 7-bit address of the target slave device. |

| SMBusData0 - RW - 8 bits - [SMBUS:05h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusData0 | 7:0 | 00h | This register should be programmed with a value to be transmitted in the data 0 field of an SMBus host interface transaction. For Block Write commands, the count of the memory should be stored in this field. The value of this register is loaded into the block transfer count field. This valid value for block command count is between 1 and 32. For block reads, count received from SMBus device is stored here. |

| SMBusData1 - RW - 8 bits - [SMBUS:06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusData1 | 7:0 | 00h | This register should be programmed with a value to be transmitted in the data 1 field of an SMBus host interface transaction. |

| SMBusBlockData - RW - 8 bits - [SMBUS:07h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusBlockData | 7:0 | 00h | This register is used to transfer data into or out of the block data storage array. |

| SMBusSlaveControl - RW - 8 bits - [SMBUS:08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SlaveEnable | 0 | 0b | Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the host controller slave port of 10h and a command field which matches the SMBus slave control register and a match of corresponding enabled events. |
| SMBusShadow1En | 1 | 0b | Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBus Shadow 1 register. |
| SMBusShadow2En | 2 | 0b | Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBus Shadow 2 register. |
| SMBusAlertEnable | 3 | 0b | Enable the generation of an interrupt or resume event on the assertion of AMBALERT_ signal. (This function is not supported). [Read-only] |
| Reserved | 7:4 | 0h | |

| SMBusShadowCmd - RW - 8 bits - [SMBUS:09h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusShadowCmd | 7:0 | 00h | This field contains the command value which was received during an external SMBus master access whose address field matched the host slave address (10h) or one of the slave shadow ports. |

| SMBusSlaveEvent - RW - 16 bits - [SMBUS:0A-0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusSlaveEvent | 15:0 | 0000h | This field contains data bits used to compare against incoming data to the SMBus Slave Data register. When a bit in this register is 1 and a corresponding bit in SMBus Slave register is set, then an interrupt or resume event is generated if the command value matches the value in the SMBus slave control register and the access was to SMBus host address 10h. |

| SlaveData - RW - 16 bits - [SMBUS:0C-0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SlaveData | 15:0 | 0000h | This field contains the data value which was transmitted during an external SMBus master access whose address field matched one of the slave shadow port addresses or the SMBus host controller slave port address of 10h. |

| SMBusTiming - RW - 8 bits - [SMBUS:0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMBusTiming | 7:0 | A0h | This register controls the frequency on the SMBUS. The formula to calculate the frequency is:<br>Frequency = 66Mhz/(SmBusTiming * 4) |

### *2.3.3 Legacy ISA and ACPI Controller*

#### *2.3.3.1 Legacy Block Registers*

There are two sets of registers in the ACPI/SMBus module. The first set is in the PCI configuration space and the registers control the behavior of the PCI interface. The second set is in the Memory/IO mapped address space. These registers control the functions of the module.

##### 2.3.3.1.1 IO-Mapped Control Registers

| Register Name | Offset Address |
|---|---|
| Dma_Ch 0 | 00h |
| Dma_Ch 1 | 02h |
| Dma_Ch 2 | 04h |
| Dma_Ch 3 | 06h |
| Dma_Status | 08h |
| Dma_WriteRequest | 09h |
| Dma_WriteMask | 0Ah |
| Dma_WriteMode | 0Bh |
| Dma_Clear | 0Ch |
| Dma_MasterClr | 0Dh |
| Dma_ClrMask | 0Eh |
| Dma_AllMask | 0Fh |
| IntrCntrlReg1 | 20h |
| IntrCntrlReg2 | 21h |
| TimerCh0 | 40h |
| TimerCh1 | 41h |
| TimerCh2 | 42h |
| Tmr1CntrlWord | 43h |
| Nmi_Status | 61h |
| RtcAddrPort | 70h |
| RtcDataPort | 71h |
| AlternatRtcAddrPort | 72h |
| AlternatRtcDataPort | 73h |
| Dma_PageCh2 | 81h |
| Dma_PageCh3 | 82h |
| Dma_PageCh1 | 83h |
| Dma_Page_Reserved1 | 84h |
| Dma_Page_Reserved2 | 85h |
| Dma_Page_Reserved3 | 86h |
| Dma_PageCh0 | 87h |
| Dma_Page_Reserved4 | 88h |
| Dma_PageCh6 | 89h |
| Dma_PageCh7 | 8Ah |
| Dma_PageCh5 | 8Bh |
| Dma_Page_Reserved5 | 8Ch |
| Dma_Page_Reserved6 | 8Dh |
| Dma_Page_Reserved7 | 8Eh |
| Dma_Refresh | 8Fh |
| FastInit | 92h |
| IntrCntrl2Reg1 | A0h |
| IntrCntrl2Reg2 | A1h |
| Dma2_Ch4Addr | C0h |
| Dma2_Ch4Cnt | C2h |
| Dma2_Ch5Addr | C4h |
| Dma2_Ch5Cnt | C6h |
| Dma2_Ch6Addr | C8h |

| Register Name | Offset Address |
|---|---|
| Dma2_Ch6Cnt | CAh |
| Dma2_Ch7Addr | CCh |
| Dma2_Ch7Cnt | CEh |
| Dma_Status | D0h |
| Dma_WriteRequest | D2h |
| Dma_WriteMask | D4h |
| Dma_WriteMode | D6h |
| Dma_Clear | D8h |
| Dma_Clear | DAh |
| Dma_ClrMask | DCh |
| Dma_ClrMask | DEh |
| NCP_Error | F0h |
| DMA1_Extend | 40Bh |
| IntrEdgeControl | 4D0h |
| DMA2_Extend | 4D6h |
| Pci_Intr_Index | C00h |
| Pci_Intr_Data | C01h |
| Pci_Error | C14h |
| CMIndex | C50h |
| CMData | C51h |
| GpmPort | C52h |
| Isa_Misc | C6Fh |
| PM2_Index | CD0h |
| PM2_Data | CD1h |
| BIOSRAM_Index | CD4h |
| BIOSRAM_Data | CD5h |
| PM_Index | CD6h |
| PM_Data | CD7h |

.

The PCI I/O registers are 32-bit registers decoded from the full 32-bit PCI address and C/BE[3:0]#. Therefore, the bytes within a 32-bit address are selected with the valid byte enables. Registers and bits within a register marked as reserved are not implemented. Writes have no effect on reserved registers. All PCI I/O registers can be accessed via 8, 16, or 32-bit cycles (i.e., each byte is individually selected by the byte enables).

| Dma_Ch 0- RW – 16 bits - [IO_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Ch 0 | 15:0 | 0000h | DMA1 Ch0 Base and Current Address |
| Dma_Ch 0 register | | | |

| Dma_Ch 1- RW – 16 bits - [IO_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Ch 1 | 15:0 | 0000h | DMA1 Ch1 Base and Current Address |
| Dma_Ch 1 register | | | |

| Dma_Ch 2- RW – 16 bits - [IO_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Ch 2 | 15:0 | 0000h | DMA2 Ch2 Base and Current Address |
| Dma_Ch 2 register | | | |

| Dma_Ch 3- RW – 16 bits - [IO_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Ch 3 | 15:0 | 0000h | DMA1 Ch3 Base and Current Address |
| Dma_Ch 3 register | | | |

| Dma_Status- RW – 8 bits - [IO_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Status | 7:0 | 00h | Returns status when read; command for write |
| Dma_Status register | | | |

| Dma_WriteRequest- RW – 8 bits - [IO_Reg: 09h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteRequest | 7:0 | 00h | Request register. |
| Dma_WriteRequest register | | | |

| Dma_WriteMask- RW – 8 bits - [IO_Reg: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteMask | 7:0 | 00h | Channel mask register. |
| Dma_WriteMask register | | | |

| Dma_WriteMode- RW – 8 bits - [IO_Reg: 0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteMode | 7:0 | 00h | Mode register. |
| Dma_WriteMode register | | | |

| Dma_Clear- RW – 8 bits - [IO_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Clear | 7:0 | 00h | Channel 0-3 DMA clear byte pointer |
| Dma_Clear register | | | |

| Dma_MasterClr- RW – 8 bits - [IO_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_MasterClr | 7:0 | 00h | Intermediate register. |
| Dma_MasterClr register | | | |

| Dma_ClrMask- RW – 8 bits - [IO_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_ClrMask | 7:0 | 00h | Channel 0-3 DMA Clear Mask |
| Dma_ClrMask register | | | |

| Dma_AllMask- RW – 8 bits - [IO_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_AllMask | 7:0 | 00h | Mask register. |
| Dma_AllMask register | | | |

| IntrCntrl1Reg1- RW – 8 bits - [IO_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IntrCntrl1Reg1 | 7:0 | 00h | IRQ0 – IRQ7:<br>Read IRR, ISR<br>Write ICW1, OCW2, OCW3 |
| IntrCntrl1Reg1register | | | |

| IntrCntrl1Reg2- RW – 8 bits - [IO_Reg: 21h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IntrCntrl1Reg2 | 7:0 | 00h | IRQ0 – IRQ7:<br>Read IMR<br>Write ICW2, ICW3, ICW4, OCW1 |
| IntrCntrl1Reg2 register | | | |

| IMCR_Index- RW – 8 bits - [IO_Reg: 22h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IMCR_Index | 7:0 | 00h | The IMCR is supported by two read/writeable IO ports 22/23h; which are used as index and data port respectively. The actual IMCR register is located at index 70h. |
| IMCR_Index register | | | |

| IMCR_Data- RW – 8 bits - [IO_Reg: 23h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IMCR_Data | 7:0 | 00h | The IMCR is supported by two read/writeable IO ports 22/23h; which are used as index and data port respectively. The actual IMCR register is located at index 70h and it is at bit 0. The actual IMCR bit can only be accessed when bit port 22 is set to 70h. Default value of IMCR is 0. |
| IMCR_Data register | | | |

| TimerCh0- RW – 8 bits - [IO_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TimerCh0 | 7:0 | 00h | 8254 Timer 1 – Counter 0 Data Port<br>This timer is known as the System Clock timer and it is always on. It is clocked internally by OSC/12 (1.19318MHz), and asserts IRQ0 every time the timer rolls over. This timer is used for time-of-day, diskette time-out, and other system timing functions. |
| TimerCh0 register | | | |

| TimerCh1- RW – 8 bits - [IO_Reg: 41h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TimerCh1 | 7:0 | 00h | 8254 Timer 1 – Counter 1 Data Port<br>This timer is normally used for ISA refresh cycles and is also clocked by OSC/12 (1.19818MHz). Since this refresh function is no longer needed (we don't have an external ISA bus), it can be used as a general purpose timing function. |
| TimerCh1 register | | | |

| TimerCh2- RW – 8 bits - [IO_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TimerCh2 | 7:0 | 00h | 8254 Timer 1 - Counter 2 Data Port<br>This is the speaker tone generator and is enabled by IO port 61H. It is clocked by OSC/12 (1.19318MHz) and directly drives the output SPKR that goes to a speaker. |
| TimerCh2 register | | | |

| Tmr1CntrlWord - RW – 8 bits - [IO_Reg: 43h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CntDownSelect | 0 | 0b | 0 – Binary countdown<br>1 – BCD countdown |
| ModeSelect | 3:1 | 000b | 000 – Asserts OUT signal at end of count<br>001 – Hardware re-triggerable one-shot<br>010 – Rate generator<br>011 – Square wave output<br>100 – Software triggered strobe<br>101 – Hardware triggered strobe<br>110 – 111 – not used |
| CmmandSelect | 5:4 | 00b | 00 – Counter latch command<br>01 – Read/write least significant byte<br>10 – Read/write most significant byte<br>11 – Read/write least, and then most significant byte |
| CounterSelect | 7:6 | 00b | 00 - Select counter 0<br>01 - Select counter 1<br>10 - Select counter 2<br>11 – Read back command |
| Tmr1CntrlWord register: This is the control word to access the 8254 timer 1. It is used to select which counter will be accessed and how it will be accessed. This register specifies the counter, the operating mode, the order and size of the count value, and whether it counts down in a 16 bit or BCD format.<br>If a counter is programmed to read or write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter; otherwise, the counter will be loaded with an incorrect value. The count must always be completely loaded with both bytes. | | | |

| Nmi_Status - RW – 8 bits - [IO_Reg: 61h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SpkrEnable | 0 | 0b | 0 - Disable counter 2<br>1 - Enable counter 2 |
| SpkrTmrEnable | 1 | 0b | 0 - Speaker timer off<br>1 - Speaker timer on |
| Parity_Nmi_En | 2 | 1b | 0 - Enable Parity Error to NMI generation (from SERR# or PERR#)<br>1 - Disable Parity Error to NMI generation and clear bit 7 |
| Iochk_Nmi_En | 3 | 1b | 0 - Enable IoChk to NMI generation<br>1 - Disable IoChk to NMI generation |
| RefClk | 4 | - | The output of the counter 1 (8254). (Read-only) |
| SpkrClk | 5 | - | The output of the counter 2. [Read-only] |
| IoChk_Nmi | 6 | - | NMI is triggered by serial IOCHK. [Read-only] |
| ParErr_Nmi | 7 | - | NMI is caused by parity error (either PERR# or SERR#). [Read-only] |
| Nmi_Status register: Independent read and write registers will be accessed at this port. When writing to port 61H, bits[3:0] allow software to enable/disable parity error NMI's and control the speaker timer. When reading port 61H, status on parity errors, speaker count, speaker control and refresh cycles is returned. | | | |

| Nmi_Enable - RW – 8 bits - [IO_Reg: 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RTC Address Port | 6:0 | 00h | This is used with either internal RTC or external RTC |
| NmiEnable | 7 | 0b | 0 - NMI enable<br>1 - NMI disable [Write-only] |
| Nmi_Enable register | | | |

| RtcDataPort - RW – 8 bits - [IO_Reg: 71h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RTC Data Port | 7:0 | 00h | This is used with either internal RTC or external RTC |

| RtcDataPort - RW – 8 bits - [IO_Reg: 71h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RtcDataPort | | | |

| AlternatRtcAddrPort - RW – 8 bits - [IO_Reg: 72h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AlternatRTCAddrPort | 7:0 | 00h | This is used with internal RTC. This port allows user to specify the full 8 bit address (instead of bank0/bank1 indexing) to access the 256 byte RTC RAM |
| AlternatRtcAddrPort | | | |

| AlternatRtcDataPort - RW – 8 bits - [IO_Reg: 73h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AlternatRTC Data Port | 7:0 | 00h | This is used with internal RTC in conjunction with port h72 |
| AlternatRtcDataPort | | | |

| Dma_PageCh2 - RW – 8 bits - [IO_Reg: 81h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh2 | 7:0 | 00h | DMA2 ch 2 page register |
| Dma_PageCh2 register | | | |

| Dma_PageCh3 - RW – 8 bits - [IO_Reg: 82h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh3 | 7:0 | 00h | DMA2 ch 3 page register |
| Dma_PageCh3 register | | | |

| Dma_PageCh1 - RW – 8 bits - [IO_Reg: 83h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh1 | 7:0 | 00h | DMA2 ch 1 page register |
| Dma_PageCh1 register | | | |

| Dma_Page_Reserved1- RW – 8 bits - [IO_Reg: 84h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved1 | 7:0 | 00h | Dma Page Reserved1 register |
| Dma_Page_Reserved1 register | | | |

| Dma_Page_Reserved2- RW – 8 bits - [IO_Reg: 85h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved2 | 7:0 | 00h | Dma Page Reserved2 register |
| Dma_Page_Reserved2 register | | | |

| Dma_Page_Reserved3- RW – 8 bits - [IO_Reg: 86h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved3 | 7:0 | 00h | Dma Page Reserved3 register |
| Dma_Page_Reserved3 register | | | |

| Dma_PageCh0 - RW – 8 bits - [IO_Reg: 87h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh0 | 7:0 | 00h | DMA2 ch 0 page register |
| Dma_PageCh0 register | | | |

| Dma_Page_Reserved4- RW – 8 bits - [IO_Reg: 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved4 | 7:0 | 00h | Dma Page Reserved4 register |
| Dma_Page_Reserved4 register | | | |

| Dma_PageCh6 - RW – 8 bits - [IO_Reg: 89h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh6 | 7:0 | 00h | DMA2 ch 6 page register |
| Dma_PageCh6 register | | | |

| Dma_PageCh7 - RW – 8 bits - [IO_Reg: 8Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh7 | 7:0 | 00h | DMA2 ch 7 page register |
| Dma_PageCh7 register | | | |

| Dma_PageCh5 - RW – 8 bits - [IO_Reg: 8Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_PageCh5 | 7:0 | 00h | DMA2 ch 5 page register |
| Dma_PageCh5 register | | | |

| Dma_Page_Reserved5- RW – 8 bits - [IO_Reg: 8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved5 | 7:0 | 00h | Dma Page Reserved5 register |
| Dma_Page_Reserved5 register | | | |

| Dma_Page_Reserved6- RW – 8 bits - [IO_Reg: 8Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved6 | 7:0 | 00h | Dma Page Reserved6 register |
| Dma_Page_Reserved6 register | | | |

| Dma_Page_Reserved7- RW – 8 bits - [IO_Reg: 8Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Page_Reserved7 | 7:0 | 00h | Dma Page Reserved7 register |
| Dma_Page_Reserved7 register | | | |

| Dma_Refresh- RW – 8 bits - [IO_Reg: 8Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Refresh | 7:0 | 00h | DMA2 ch4 page register. |
| Dma_Refresh register | | | |

| FastInit- RW – 8 bits - [IO_Reg: 92h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FastInit | 0 | 0b | FAST_INIT. This read/write bit provides a fast software executed processor reset function. Writing<br>a 1 to this bit will cause the INIT assertion for approximately 4ms.<br>Before another INIT pulse can be generated via this register, this bit must be written back to a 0. |
| A20EnB | 1 | 0b | A20Enable Bar bit; if set to 1, A20M is disabled. |

| IntrCntrl2Reg1- RW – 8 bits - [IO_Reg: A0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IntrCntrl2Reg1 | 7:0 | 00h | IRQ8 – IRQ15:<br>Read IRR, ISR<br>Write ICW1, OCW2, OCW3 |
| IntrCntrl2Reg1 register | | | |

| IntrCntrl2Reg2- RW – 8 bits - [IO_Reg: A1h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IntrCntrl2Reg2 | 7:0 | 00h | IRQ8 – IRQ15:<br>Read IMR<br>Write ICW2, ICW3, ICW4, OCW1 |
| IntrCntrl2Reg2 register | | | |

| Dma2_Ch4Addr - RW – 8 bits - [IO_Reg: C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch4Addr | 7:0 | 00h | DMA2 Ch4 Base and Current Address |
| Dma2_Ch4Addr register | | | |

| Dma2_Ch4Cnt – RW – 8 bits - [IO_Reg: C2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch4Cnt | 7:0 | 00h | DMA2 Ch4 Base and Current Count |
| Dma2_Ch4Cnt register | | | |

| Dma2_Ch5Addr - RW – 8 bits - [IO_Reg: C4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch5Addr | 7:0 | 00h | DMA2 Ch5 Base and Current Address |
| Dma2_Ch5Addr register | | | |

| Dma2_Ch5Cnt - RW – 8 bits - [IO_Reg: C6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch5Cnt | 7:0 | 00h | DMA2 Ch4 Base and Current Count |
| Dma2_Ch5Cnt register | | | |

| Dma2_Ch6Addr - RW – 8 bits - [IO_Reg: C8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch6Addr | 7:0 | 00h | DMA2 Ch6 Base and Current Address |
| Dma2_Ch6Addr register | | | |

| Dma2_Ch6Cnt - RW – 8 bits - [IO_Reg: CAh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch6Cnt | 7:0 | 00h | DMA2 Ch6 Base and Current Count |
| Dma2_Ch6Cnt register | | | |

| Dma2_Ch7Addr - RW – 8 bits - [IO_Reg: CCh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch7Addr | 7:0 | 00h | DMA2 Ch5 Base and Current Address |
| Dma2_Ch7Addr register | | | |

| Dma_Ch7Cnt - RW – 8 bits - [IO_Reg: CEh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma2_Ch7Cnt | 7:0 | 00h | Channel 7 DMA base and current count |
| Dma2_Ch7Cnt register | | | |

| Dma_Status - RW – 8 bits - [IO_Reg: D0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Status | 7:0 | 00h | DMA2 status register |
| Dma_Status register | | | |

| Dma_WriteRequest - RW – 8 bits - [IO_Reg: D2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteRequest | 7:0 | 00h | DMA2 request register |
| Dma_WriteRequest register | | | |

| Dma_WriteMask - RW – 8 bits - [IO_Reg: D4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteMask | 7:0 | 00h | DMA2 channel mask register |
| Dma_WriteMask register | | | |

| Dma_WriteMode - RW – 8 bits - [IO_Reg: D6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_WriteMode | 7:0 | 00h | DMA2 mode register |
| Dma_WriteMode register | | | |

| Dma_Clear - RW – 8 bits - [IO_Reg: D8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Clear | 7:0 | 00h | Channel 4-7 clear byte pointer |
| Dma_Clear register | | | |

| Dma_Clear - RW – 8 bits - [IO_Reg: DAh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_Clear | 7:0 | 00h | Channel 4-7 DMA master clear |
| Dma_Clear register | | | |

| Dma_ClrMask - RW – 8 bits - [IO_Reg: DCh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_ClrMask | 7:0 | 00h | Channel 4-7 DMA Clear Mask |
| Dma_ClrMask register | | | |

| Dma_ClrMask - RW – 8 bits - [IO_Reg: DEh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Dma_AllMask | 7:0 | 00h | DMA2 mask register |
| Dma_AllMask register | | | |

| NCP_Error - RW – 8 bits - [IO_Reg: F0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 6:0 | 00h | |
| WarmBoot | 7 | 0b | Warm or cold boot indicator<br>0 – Cold<br>1 – Warm, this bit is set when any value is written to this register; |

| NCP_Error - RW – 8 bits - [IO_Reg: F0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NCP_Error register: In addition to the WarmBoot function, writing to this port will assert IGNNE# if FERR# is true. If FERR# is false, then write to this port will not assert IGNNE#. | | | |

| DMA1_Extend - RW – 8 bits - [IO_Reg: 40Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA1_Extend | 7:0 | 00h | DMA1 extended write mode register |
| DMA1_Extend register | | | |

| IntrEdgeControl- RW – 16 bits - [IO_Reg: 4D0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IRQ0Control | 0 | 0b | 1 = Level, 0 = Edge |
| IRQ1Control | 1 | 0b | 1 = Level, 0 = Edge |
| Reserved | 2 | 0b | |
| IRQ3Control | 3 | 0b | 1 = Level, 0 = Edge |
| IRQ4Control | 4 | 0b | 1 = Level, 0 = Edge |
| IRQ5Control | 5 | 0b | 1 = Level, 0 = Edge |
| IRQ6Control | 6 | 0b | 1 = Level, 0 = Edge |
| IRQ7Control | 7 | 0b | 1 = Level, 0 = Edge |
| IRQ8Control | 8 | 0b | (Read Only) Always edge |
| IRQ9Control | 9 | 0b | 1 = Level, 0 = Edge |
| IRQ10Control | 10 | 0b | 1 = Level, 0 = Edge |
| IRQ11Control | 11 | 0b | 1 = Level, 0 = Edge |
| IRQ12Control | 12 | 0b | 1 = Level, 0 = Edge |
| Reserved | 13 | 0b | |
| IRQ14Control | 14 | 0b | 1 = Level, 0 = Edge |
| IRQ15Control | 15 | 0b | 1 = Level, 0 = Edge |
| IntrEdgeControl register: This register programs each interrupt to be either edge or level sensitive. | | | |

| DMA2_Extend - RW – 8 bits - [IO_Reg: 4D6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA2_Extend | 7:0 | 00h | DMA2 extended write mode register |
| DMA2_Extend register | | | |

| Pci_Intr_Index - RW – 8 bits - [IO_Reg: C00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Pci_Intr_Index | 7:0 | 00h | PCI interrupt index – selects which PCI interrupt to map<br>0h – INTA#<br>1h – INTB#<br>2h – INTC#<br>3h – INTD#<br>4h – Interrupt generated by ACPI<br>5h – Interrupt generated by Sm Bus<br>6h – Reserved<br>7h – Ac97 audio<br>8h – Ac97 modem<br>9h – INTE#<br>Ah – INTF#<br>Bh – INTG#<br>Ch – INTH# |
| Pci_Intr_Index register | | | |

| Pci_Intr_Data - RW – 8 bits - [IO_Reg: C01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Pci_Intr_Data | 7:0 | 00h | PCI redirection register; map PCI interrupt addressed by Pci_Intr_Index to a PIC IRQ<br>[7:4] – Not used<br>Straight encoding of [3:0] to : IRQ0 thru IRQ15<br>Note: Do not map to IRQ 0, 2, 8, 13 – they are reserved |
| Pci_Intr_Data register<br>**Note:** If IOXAPIC is enabled, software must make sure interrupts are not re-routed; i.e., they should all be set to 0. When IOXAPIC is enabled, IRQ[15:0] are routed directly to INT[15:0]. INTH#, INTG#, INTF#, INTE#, INTD#, INTC#, INTB#, and INTA# are routed to INTIN[23:16], SMBus interrupt is routed to INTIN[20], SCI# is routed to INTIN[9], and the High Precision Event Timer (also called the Multimedia Timer) can be routed to either INTIN[22] or INTIN[23]. | | | |

| Pci_Error - RW – 8 bits - [IO_Reg: C14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Serr_Nmi_Status | 0 | - | Set to 1 when NMI generation is enabled and SERR# has been asserted due to a PCI error. Cleared by writing a one to port 61h, bit 2. [Read-only] |
| Perr_Nmi_Status | 1 | - | Set to 1 when NMI generation is enabled and PERR# has been asserted due to a PCI data parity error. Cleared by writing a one to port 61h, bit 2. [Read-only] |
| Serr_Nmi | 2 | 1b | Enable NMI generation from SERR#<br>0 – Enable<br>1 – Disable |
| Perr_Nmi | 3 | 1b | Enable NMI generation from PERR#<br>0 – Enable<br>1 – Disable |
| Reserved | 7:4 | 0h | |
| Pci_Error register | | | |

| CMIndex - RW – 8 bits - [IO_Reg: C50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CMIndex | 7:0 | 00h | Index register to client management register block<br>00h – IdRegister<br>02h – TempStatus<br>03h – TempInterrupt<br>12h – SmBus control (control to Gpoc[3:0] pins thru Bit Bang)<br>13h - Misccontrol<br>others – super IO – not used |
| CMIndex register | | | |

| CMData - RW – 8 bits - [IO_Reg: C51h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CMData | 7:0 | 00h | Data register to client management register block |
| CMData register | | | |

| GpmPort - RW – 8 bits - [IO_Reg: C52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpm | 7:0 | -- | If CMIndex.13h[7:6] == 00, then this is the read port for GPM[7:0].<br>If CMIndex.13h[7:6]= 01, then this is the output enable for GPM[7:0], 0=enable, 1=tristate<br>If CMIndex.13h[7:6]=10, then this is the output state control (providing enable is turned on) |

| GpmPort - RW – 8 bits - [IO_Reg: C52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GpmPort register | | | |

| Isa_Misc - RW – 8 bits - [IO_Reg: C6Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 5:0 | 00h | |
| FlashRomEn | 6 | 0b | Flash program enable. This only applies if the chip is strapped to have the ROM on the PCI bus . Set to 1 to enable programming of the Flash ROM. During write cycles to the system ROM space, ROMCS# is only asserted if this bit is set 1. |
| Reserved | 7 | 0b | |
| Isa_Misc register: FlashRom Enable. This only applies if the chip is strapped to have the ROM on the secondary PCI bus. | | | |

| BIOSRAM_Index - RW – 8 bits - [IO_Reg: CD4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BiosRamIndex | 7:0 | 00h | BIOS RAM index register. This register selects one of the 256 bytes of BIOS RAM. Data in this RAM is preserved until RSMRST# is asserted, or S5 power is lost.) |
| BiosRamIndex register | | | |

| BIOSRAM_Data - RW – 8 bits - [IO_Reg: CD5h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BiosRamData | 7:0 | 00h | Power management data register. This register provides the read/write access to the indexed register. |
| BiosRamData register | | | |

| PM_Index - RW – 8 bits - [IO_Reg: CD6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PM_Index | 7:0 | 00h | Power management index register. This register selects one of the Power Management registers. (See section *2.3.3.2* for more information.) |
| PM_Index register | | | |

| PM_Data - RW – 8 bits - [IO_Reg: CD7h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PM_Data | 7:0 | 00h | Power management data register. This register provides the read/write access to the indexed register. (See section *2.3.3.2* for more information.) |
| PM_Data register | | | |

#### 2.3.3.1.2 Client Management Registers (Accessed through C50h and C51h)

| Register Name | Offset Address |
|---|---|
| IdRegister | 00h |
| TempStatus | 02h |
| TempInterrupt | 03h |
| I2Ccontrol | 12h |
| Index13 | 13h |

| IdRegister - R – 8 bits - [IO_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IdRegister | 7:0 | 00h | |
| IdRegister register | | | |

| TempStatus - R – 8 bits - [IO_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TALERT | 0 | - | Logical status of TALERT/GPIO64 input.<br>Read will clear this bit. |
| Reserved | 7:1 | 00h | |
| TempStatus register | | | |

| TempInterrupt - RW – 8 bits - [IO_Reg: 03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| TempSmiEnable | 1 | 0b | 1 - Generate SMI# upon TALERT<br>0 - Do not generate SMI# upon TALERT |
| Reserved | 2 | 0b | |
| ScratchBit | 3 | 0b | This placebo bit has no function, but it may be used for software status |
| Reserved | 7:4 | 0h | |
| TempInterrupt register | | | |

| I2CControl - RW – 8 bits - [IO_Reg: 12h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpoc0Status | 0 | - | Gpoc0 status. [Read-only] |
| Gpoc1Status | 1 | - | Gpoc1 status. [Read-only] |
| Gpoc0_OE_ | 2 | 1b | 1 - Gpoc0 is tristate<br>0 - Gpoc0 is asserted low |
| Gpoc1_OE_ | 3 | 1b | 1 - Gpoc1 is tristate<br>0 - Gpoc1 is asserted low |
| Gpoc2Status | 4 | - | Gpoc2 status. [Read-only] |
| Gpoc3Status | 5 | - | Gpoc3 status. [Read-only] |
| Gpoc2_OE_ | 6 | 1b | 1 - Gpoc2 is tristate<br>0 - Gpoc2 is output enabled |
| Gpoc3_OE_ | 7 | 1b | 1 – Gpoc3 is tristate<br>0 – Gpoc3 is output enabled |
| I2CControl register: Writing to this register controls the output of the four Gpoc outputs (GPOC[3:0]_OE) and reading returns the status of these pins. | | | |

| Index13– 8 bits - [IO_Reg: 13h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPIO64En | 0 | 0b | GPIO64 output enable<br>1 = Enable<br>0 = Tristate |
| GPIO64OUT | 1 | 0b | GPIO64 output data |
| Gpoc[3:2]OUT | 3:2 | 00b | Gpoc[3:2] output data |
| ScratchBit | 5:4 | 00b | Placebo bit, has no function but may be used for software status |
| GpmPortSel | 7:6 | 00b | 00-Read port<br>01-Output enable<br>10-Output port |
| Index13 misc control register | | | |

#### 2.3.3.1.3 System Reset Register (IO CF9)

Note: Refer to PM IO reg x85 for a detailed description. This register has been designed to be dual-port accessible.

### *2.3.3.2 Power Management (PM) Registers*

The power management (PM) block is resident in the PCI/LPC/ISA bridge. The PM registers are accessed via IO mapped registers xCD6h and xCD7h. The index address is first programmed into IO register xCD6h. Read or write values are accessed through IO register xCD7h.

| Register Name | Offset Address |
|---|---|
| MiscControl | 00h |
| MiscStatus | 01h |
| SmiWakeUpEventEnable1 | 02h |
| SmiWakeUpEventEnable2 | 03h |
| SmiWakeUpEventEnable3 | 04h |
| SmiWakeUpEventStatus1 | 05h |
| SmiWakeUpEventStatus2 | 06h |
| SmiWakeUpEventStatus3 | 07h |
| InactiveTmrEventEnable1 | 08h |
| InactiveTmrEventEnable2 | 09h |
| InactiveTmrEventEnable3 | 0Ah |
| PmTmr1InitValue | 0Bh |
| PmTmr1CurValue | 0Ch |
| PwrLedExtEvent | 0Dh |
| PwrLedExtEvent | 0Eh |
| AcpiStatus | 0Fh |
| AcpiEn | 10h |
| S1AgpStpEn | 11h |
| PmTmr2InitValue | 12h |
| PmTmr2CurValue | 13h |
| ProgramIo0RangeLo | 14h |
| ProgramIo0RangeHi | 15h |
| ProgramIo1RangeLo | 16h |
| ProgramIo1RangeHi | 17h |
| ProgramIo2RangeLo | 18h |
| ProgramIo2RangeHi | 19h |
| ProgramIo3RangeLo | 1Ah |
| ProgramIo3RangeHi | 1Bh |
| ProgramIoEnable | 1Ch |
| IOMonitorStatus | 1Dh |
| InactiveTmrEventEnable4 | 1Eh |
| AcpiPm1EvtBlkLo | 20h |
| AcpiPm1EvtBlkHi | 21h |
| AcpiPm1CntBlkLo | 22h |
| AcpiPm1CntBlkHi | 23h |
| AcpiPmTmrBlkLo | 24h |
| AcpiPmTmrBlkHi | 25h |
| CpuControlLo | 26h |
| CpuControlHi | 27h |
| AcpiGpe0BlkLo | 28h |
| AcpiGpe0BlkHi | 29h |
| AcpiSmiCmdLo | 2Ah |
| AcpiSmiCmdHi | 2Bh |

| Register Name | Offset Address |
|---|---|
| AcpiPmaCntBlkLo | 2Ch |
| AcpiPmaCntBlkHi | 2Dh |
| AcpiSsCntBlkLo | 2Eh |
| AcpiSsCntBlkHi | 2Fh |
| GEvtConfig0 | 30h |
| GEvtConfig1 | 31h |
| GPMConfig0 | 32h |
| GPMConfig1 | 33h |
| GPMConfig2 | 34h |
| GPMConfig3 | 35h |
| GEvtLevelConfig | 36h |
| GPMLevelConfig0 | 37h |
| GPMLevelConfig1 | 38h |
| GEvtStatus | 39h |
| PMEStatus0 | 3Ah |
| PMEStatus1 | 3Bh |
| OthersConfig | 3Ch |
| VRT_T1 | 3Eh |
| VRT_T2 | 3Fh |
| AD_Pull_UpB | 40h |
| PM_Enable | 50h |
| TPRESET1 | 51h |
| TPRESET2 | 52h |
| TESTENABLE | 53h |
| PWRBTTN_CLR | 54h |
| SoftPciRst | 55h |
| Reserved | 56h |
| Reserved | 57h |
| Reserved | 58h |
| MiscOption | 59h |
| SmiSciSts0 | 5Ah |
| SmiSciSts1 | 5Bh |
| SmiSciSts2 | 5Ch |
| SmiSciSts3 | 5Dh |
| MwaitEnable | 5Eh |
| MwaitSmiSts | 5Fh |
| Options_0 | 60h |
| Options_1 | 61h |
| Shadow_SCI | 62h |
| SwitchVoltageTime | 63h |
| SwitchGHI_Time | 64h |
| UsbPMControl | 65h |
| MiscEnable66 | 66h |
| MiscEnable67 | 67h |
| MiscEnable68 | 68h |
| WatchDogTimerControl | 69h |
| WatchDogTimerBase0 | 6Ch |
| WatchDogTimerBase1 | 6Dh |
| WatchDogTimerBase2 | 6Eh |
| WatchDogTimerBase3 | 6Fh |
| S_LdtStartTime | 70h |
| EnhanceOption | 71h |
| C4Control | 72h |
| PopUpEndTime | 73h |
| PwrFailShadow | 74h |
| Tpreset1b | 75h |
| SOS3ToS5Enable0 | 76h |
| SOS3ToS5Enable1 | 77h |

| Register Name | Offset Address |
|---|---|
| SOS3ToS5Enable2 | 78h |
| SOS3ToS5Enable3 | 79h |
| NoStatusControl0 | 7Ah |
| NoStatusControl1 | 7Bh |
| MiscEnable7C | 7Ch |
| DprSlpVrMinTime | 7Dh |
| SMAF0 | 80h |
| SMAF1 | 81h |
| SMAF2 | 82h |
| SMAF3 | 83h |
| WakePinCntl | 84h |
| CF9Rst | 85h |
| ThermThrotCntl | 86h |
| LdtStpCmd | 87h |
| LdtStartTime | 88h |
| AgpStartTime | 89h |
| LdtAgpTimeCntl | 8Ah |
| StutterTime | 8Bh |
| StpClkDlyTime | 8Ch |
| AbPmeCntl | 8Dh |
| FakeAsr | 8Eh |
| FakeAsrEn | 8Fh |
| GEVENTOUT | 90h |
| GEVENTEN | 91h |
| GEVENTIN | 92h |
| GPM98OUT | 94h |
| GPM98EN | 95h |
| GPM98IN | 96h |
| K8C1ePort | 99:98h |
| EnhanceControl | 9Ah |
| K8C1eReadPort | 9Bh |
| MsiSignature | 9E:9Ch |
| AutoArbDisWaitTime | 9Fh |
| Programlo4RangeLo | A0h |
| Programlo4RangeHi | A1h |
| Programlo5RangeLo | A2h |
| Programlo5RangeHi | A3h |
| Programlo6RangeLo | A4h |
| Programlo6RangeHi | A5h |
| Programlo7RangeLo | A6h |
| Programlo7RangeHi | A7h |
| PIO7654Enable | A8h |
| PIO7654Status | A9h |
| PllDebug | B1:B0h |
| AltDebugBusCntrl | B2h |
| C2Count | B3h |
| C3Count | B4h |

| MiscControl - RW – 8 bits - [PM_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| Timer1ExpEn | 1 | 0b | Set to 1 to enable SMI# when PM_TIMER1 expires. When PM_TIMER1 (inactivity) expires, the SB will update bit 1 of MiscStatus and issue SMI#. This bit allows the software to disable/enable all inactivity timer reload enables at indexes 08,09, and 0A. |
| Timer2ExpEn | 2 | 0b | Set to 1 to enable SMI# when PM_TIMER2 expires. |

| MiscControl - RW – 8 bits - [PM_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3 | 0b | |
| SmiReq | 4 | 0b | Software initiated SMI#. When set, SB will update bit [4] of the MiscStatus and issue SMI#. |
| Reserved | 7:5 | 000b | |
| MiscControl register | | | |

| MiscStatus - RW – 8 bits - [PM_Reg: 01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiEvent | 0 | 0b | SB sets this bit to indicate an SMI# was issued due to events specified by index 02, 03, 04, 1C, or A8h |
| Timer1Exp | 1 | 0b | SB sets this bit to indicate that PM_TIMER1 has expired. |
| Timer2Exp | 2 | 0b | SB sets this bit to indicate that PM_TIMER2 has expired. |
| Reserved | 3 | 0b | |
| SmiReq | 4 | 0b | SB sets this bit to indicate the software initiated SMI# was issued. |
| Reserved | 6:5 | 00b | |
| StatusRst | 7 | 0b | Writing a 1 to this location will reset PM status registers 05h, 06h, and 07h, 1Dh, and A9h. This mechanism provides a quick way to reset all status. |
| MiscStatus register | | | |

| SmiWakeUpEventEnable1 - RW – 8 bits - [PM_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiWakeUpEventEnable1 | 7:0 | 00h | Enable SMI# on IRQ[15:8] activity. |
| SmiWakeUpEventEnable1 register. | | | |

| SmiWakeUpEventEnable2 - RW – 8 bits - [PM_Reg: 03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiWakeUpEventEnable2 | 7:0 | 00h | Enables SMI# on {IRQ[7:3], NMI, IRQ[1:0]} activity. |
| SmiWakeUpEventEnable2 register. | | | |

| SmiWakeUpEventEnable3 - RW – 8 bits - [PM_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0 | 0 | 0b | Enables SMI# on external event input 0 |
| ExtEvent1 | 1 | 0b | Enables SMI# on external event input 1 |
| GAME_SMI_EN | 2 | 0b | Enables SMI# on game port activity (201h) |
| FDD_SMI_EN | 3 | 0b | Enables SMI# on floppy drive activity |
| HDD_SMI_EN | 4 | 0b | Enables SMI# on IDE device activity (201h) |
| COM_SMI_EN | 5 | 0b | Enables SMI# on serial ports activity (201h) |
| LPT_SMI_EN | 6 | 0b | Enables SMI# on parallel port activity (201h) |
| SLP_SMI_EN | 7 | 0b | Enables SMI# on sleep command |
| SmiWakeUpEventEnable3 register. | | | |

| SmiWakeUpEventStatus1 - RW – 8 bits - [PM_Reg: 05h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiWakeUpEventStatus1 | 7:0 | 00h | Set to 1 to identify IRQ[15:8] activity as source of SMI#. |
| SmiWakeUpEventStatus1 register. | | | |

| SmiWakeUpEventStatus2 - RW – 8 bits - [PM_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiWakeUpEventStatus2 | 7:0 | 00h | Set to 1 to identify IRQ[7:0] activity as source of SMI#. |

| SmiWakeUpEventStatus2 - RW – 8 bits - [PM_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiWakeUpEventStatus2 register. | | | |

| SmiWakeUpEventStatus3 - RW – 8 bits - [PM_Reg: 07h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvt0_SMI_Status | 0 | 0b | Set to 1 to indicate ExtEvent0 as the source of SMI#. Write 1 to clear this bit. |
| ExtEvt1_SMI_Status | 1 | 0b | Set to 1 to indicate ExtEvent1 as the source of SMI#. Write 1 to clear this bit. |
| GAME_SMI_Status | 2 | 0b | Set to 1 to indicate game port activity as the source of SMI#. Write 1 to clear this bit. |
| FDD_SMI_Status | 3 | 0b | Set to 1 to indicate floppy activity as the source of SMI#. Write 1 to clear this bit. |
| HDD_SMI_Status | 4 | 0b | Set to 1 to indicate IDE activity as the source of SMI#. Write 1 to clear this bit. |
| COM_SMI_Status | 5 | 0b | Set to 1 to indicate serial port activity as the source of SMI#. Write 1 to clear this bit. |
| LPT_SMI_Status | 6 | 0b | Set to 1 to indicate parallel port activity as the source of SMI#. Write 1 to clear this bit. |
| SLP_SMI_Status | 7 | 0b | Set to 1 to indicate sleep command as the source of SMI#. Write 1 to clear this bit. |
| SmiWakeUpEventStatus3 register: Set to one to identify PioRng. | | | |

| InactiveTmrEventEnable1 - RW – 8 bits - PM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| InactiveTmrEventEnable1 | 7:0 | 00h | Enables Timer 1 reload on IRQ[15:8] activity. |
| InactiveTmrEventEnable1 register. | | | |

| InactiveTmrEventEnable2 - RW – 8 bits - [PM_Reg: 09h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| InactiveTmrEventEnable2 | 7:0 | 00h | Enables Timer 1 reload on {IRQ[7:3], NMI, IRQ[1:0]} activity. |
| InactiveTmrEventEnable2 register. | | | |

| InactiveTmrEventEnable3 - RW – 8 bits - [PM_Reg: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvt0_Timer1Enable | 0 | 0b | Enables Timer1 reload on ExtEvent0 inactivity |
| ExtEvt1_Timer1Enable | 1 | 0b | Enables Timer1 reload on ExtEvent1 inactivity |
| GamePort_Timer1Enable | 2 | 0b | Enables Timer1 reload on Gameport inactivity |
| Floppy_Timer1Enable | 3 | 0b | Enables Timer1 reload on Floppy port inactivity |
| IDE_Timer1Enable | 4 | 0b | Enables Timer1 reload on IDE port inactivity |
| COM_Timer1Enable | 5 | 0b | Enables Timer1 reload on COM port inactivity |
| Parallel_Timer1Enable | 6 | 0b | Enables Timer1 reload on Parallel port inactivity |
| Reserved | 7 | 0b | |
| InactiveTmrEventEnable3 register. | | | |

| PmTmr1InitValue - RW – 8 bits - [PM_Reg: 0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PmTmr1InitValue | 5:0 | 000000b | 6 bit-timer; Initial/reload value for 6 bit decrementing counter. Count range from 1 minute to 64 minutes with 4 second accuracy. |
| Reserved | 7:6 | 00b | |

| PmTmr1CurValue - R – 8 bits - [PM_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PmTmr1CurValue | 5:0 | - | Current value of decrementing counter |
| Reserved | 7:6 | 00b | |

| PwrLedExtEvent - RW – 8 bits - [PM_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0State | 0 | - | Logical value of EXTEVENT0. [Read-only] |
| ExtEvent1State | 1 | - | Logical value of EXTEVENT1. [Read-only] |
| ExtEvent0RF | 2 | 0b | Set to one to configure EXTEVENT0 as rising edge sensitive. 0 as falling edge sensitive |
| ExtEvent1RF | 3 | 0b | Set to one to configure EXTEVENT1 as rising edge sensitive. 0 as falling edge sensitive |
| Reserved | 7:4 | 0100b | |
| PwrLedExtEvent register. | | | |

| AcpiControl - RW – 8 bits - [PM_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiSmiEn | 0 | 0b | Set to 1 to enable SMI# generation when ACPI driver writes one to GBL_RLS (bit [2] of AcpiPm1CntBlk:00h). When an SMI# is issued, the SB updates bit [0] of the AcpiStatus register. |
| BIOS_RLS | 1 | 0b | Writing 1 to this bit will cause GblStatus to be set. This bit is always read back as 0. |
| SmiCmdEn | 2 | 0b | Set to 1 to enable SMI# generation when ACPI driver writes to the SmiCmd port. When set to 1, SB will update bit 2 of MiscStatus and issue an SMI# on write to SmiCmd port. |
| AcpiDecodeEnable | 3 | 0b | Set to 1 to enable SB to decode the ACPI I/O address space. When set, SB uses the contents of the PM registers at index 20-2B to decode ACPI I/O address. |
| VRT_SMI_En | 4 | 0b | Set to 1 to enable SB to generate SMI# upon (RTC) VRT low. |
| Reserved | 5 | 0b | |
| RtcClkPUB | 6 | 0b | This bit controls the integrated pull-up for RTCCLK<br>0: Enable<br>1: Disable |
| BG1RESDIV0_SEL | 7 | 0b | This bit controls the CPU receiver Vref<br>0: Vref = Vcpu / 2 (recommended setting for P4 CPU)<br>1: Vref = Vbandgap / 2 = 0.6v (recommended setting for K8 CPU) |
| AcpiControl register. | | | |

| AcpiStatus- R – 8 bits - PM_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiSmiStatus | 0 | - | Set to 1 by SB to indicate SMI# was due to write to Acpi power management register. |
| SerrSmiStatus | 1 | - | SMI# due to SERR# |
| SmiCmdStatus | 2 | - | Set to one by SB to indicate SMI# was due to write to AcpiSmiCmd port. |
| SmSmiStatus | 3 | - | SmBus SMI# status |
| UsbSmiStatus | 4 | - | USB SMI# status |
| SerSmiStatus | 5 | - | Serial SMI# status |
| RtcAvailable | 6 | - | 1: Rtc clock running<br>0: Bad Rtc clock. RTC battery may not be present |
| NbThermStatus | 7 | - | NB thermal event status |

| AcpiEn - RW – 8 bits - [PM_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| EOSEnale | 0 | 0b | Set 1 by software and clear by hardware. This bit needs to be set in order to generate SMI#/SCI |
| RTC_En_En | 1 | 0b | RTC_EN enable bit. When this bit is set, RTC_EN (AcpiPmEvtBlk, index x02, bit 10) is visible; otherwise, it is always 0 |
| Reserved | 2 | 0b | |
| TMR_En_En | 3 | 0b | TMR_EN enable bit. When this bit is set, TM_EN (AcpiPmEvtBlk, index x02, bit 0) is visible; otherwise, it is always 0 |
| LEVENT_EN_EN | 4 | 0b | LEVENT_EN enable bit. When this bit is set, LEVENT_EN (AcpiGpe0Blk, index x04, bit 8) is visible; otherwise, it is always 0 |
| GBL_EN_EN | 5 | 0b | GBL_EN enable bit. When this bit set, GBL_EN (AcpiPmEvtBlk, index x02, bit 5) is visible; otherwise, it is always 0 |
| PciExpWakeDisEn | 6 | 0b | PciExpWakeDis enable bit. When this bit is set, PciExpWakeDis (AcpiPmEvtBlk, index x02, bit 14) is visible; otherwise, it is always 1 |
| GpioToGEventSel | 7 | 0b | This bit is applicable to ASIC Revision A12 and above.<br>0: ACPI_EVENT[30] register is routed to use GPIO2 as input<br>1: ACPI_EVENT[30] register is routed to use GPIO66 as input |

| S1AgpStpEn - RW – 8 bits - [PM_Reg: 11h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S1AgpStpEn | 0 | 0b | AgpStp enable in S1 state<br>1: Enable<br>0: Disable |
| Reserved | 7:1 | 00h | |
| S1AgpStpEn register. | | | |

| PmTmr2InitValue - RW – 8 bits - [PM_Reg: 12h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PmTmr2InitValue | 7:0 | 00h | PmTmr2 load value |
| PmTmr2InitValue register. | | | |

| PmTmr2CurValue - R – 8 bits - [PM_Reg: 13h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PmTmr2CurValue | 7:0 | -- | PmTmr2 current value |
| PmTmr2CurValue register. | | | |

| ProgramIo0RangeLo - RW – 8 bits - [PM_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo0Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits [2:0]. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo0RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo0RangeLo register. | | | |

| ProgramIo0RangeHi - RW – 8 bits - [PM_Reg: 15h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo0RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo0RangeHi register. | | | |

| ProgramIo1RangeLo - RW – 8 bits - [PM_Reg: 16h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo1Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies to the other three bits [2:0]. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramI01RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo1RangeLo register. | | | |

| ProgramIo1RangeHi - RW – 8 bits - [PM_Reg: 17h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo1RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIO1RangeHi register. | | | |

| ProgramIo2RangeLo - RW – 8 bits - [PM_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo2Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo2RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |

| ProgramIo2RangeLo - RW – 8 bits - [PM_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo2RangeLo register. | | | |

| ProgramIo2RangeHi - RW – 8 bits - [PM_Reg: 19h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo2RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo2RangeHi register. | | | |

| ProgramIo3RangeLo - RW – 8 bits - [PM_Reg: 1Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo3Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies to the other three bits [2:0]. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo3RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo3RangeLo register. | | | |

| ProgramIo3RangeHi - RW – 8 bits - [PM_Reg: 1Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo3RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo3RangeHi register. | | | |

| ProgramIoEnable - RW – 8 bits - [PM_Reg: 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AD_LIB_MonitorEn | 0 | 0b | Enables AD_LIB monitoring to trigger SMI#, 1=On, 0=Off |
| MIDI_MonitorEn | 1 | 0b | Enables MIDI monitoring to trigger SMI#; 1= On, 0= Off |
| AudioMSSMonitorEn | 2 | 0b | Enables Audio/MSS monitoring to trigger SMI#; 1= On, 0 = Off |
| MouseKbMonitorEn | 3 | 0b | Enables Mouse/Keyboard monitoring to trigger SMI#; 1= On, 0 = Off |
| ProgramIo3Enable | 4 | 0b | Enables IO monitoring for ProgramIO3 (defined by index 1A, 1B); 1= On, 0 = Off |
| ProgramIo2Enable | 5 | 0b | Enables IO monitoring for ProgramIO2 (defined by index 18, 19); 1= On, 0 = Off |
| ProgramIo1Enable | 6 | 0b | Enables IO monitoring for ProgramIO1 (defined by index 16, 17); 1= On, 0 = Off |
| ProgramIo0Enable | 7 | 0b | Enables IO monitoring for ProgramIo0 (defined by index 14, 15); 1= On, 0 = Off |
| IOMonitorEn register | | | |

| IOMonitorStatus - RW – 8 bits - [PM_Reg: 1Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AD_LIB_MonitorStatus | 0 | - | AD_LIB status bit; write 1'b1 to clear the status |
| MIDI_MonitorStatus | 1 | - | MIDI status bit; write 1'b1 to clear the status bit |
| AudioMSSMonitorStatus | 2 | - | Audio/MSS status bit; write 1'b1 to clear the status bit |

| IOMonitorStatus - RW – 8 bits - [PM_Reg: 1Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MouseKbMonitorStatus | 3 | - | Mouse/keyboard status bit; write 1'b1 to clear the status bit |
| ProgramIo3Status | 4 | - | Programmable IO 3 status bit; write 1'b1 to clear the status bit |
| ProgramIo2Status | 5 | - | Programmable IO 2 status bit; write 1'b1 to clear the status bit |
| ProgramIo1Status | 6 | - | Programmable IO 1 status bit; write 1'b1 to clear the status bit |
| ProgramIo0Status | 7 | - | Programmable IO 0 status bit; write 1'b1 to clear the status bit |
| IOMonitorStatus register | | | |

| InactiveTmrEventEnable4 - RW – 8 bits - [PM_Reg: 1Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AD_LIB_Timer1Enable | 0 | 0b | Enables Timer1 reload on AD_LIB inactivity |
| MIDI_Timer1Enable | 1 | 0b | Enables Timer1 reload on MIDI inactivity |
| Audio_Timer1Enable | 2 | 0b | Enables Timer1 reload on Audio inactivity |
| Keyboard_Time1Enable | 3 | 0b | Enables Timer1 reload on Keyboard/Mouse port inactivity |
| PIO3_Timer1Enable | 4 | 0b | Enables Timer1 reload on PIO3 port inactivity |
| PIO2_Timer1Enable | 5 | 0b | Enables Timer1 reload on PIO2 port inactivity |
| PIO1_Timer1Enable | 6 | 0b | Enables Timer1 reload on PIO1 port inactivity |
| PIO0_Timer1Enable | 7 | 0b | Enables Timer1 reload on PIO0 port inactivity |
| InactiveTmrEventEnable4 register. | | | |

| AcpiPm1EvtBlkLo - RW – 8 bits - [PM_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 00b | |
| AcpiPm1EvtBlkLo | 7:2 | 00h | These bits define the least significant byte of the 16 bit I/O range base address of the ACPI power management Event Block. Bit 2 corresponds to Addr[2] and bit 7 corresponds to Addr[7]. |
| AcpiPm1EvtBlkLo register. | | | |

| AcpiPm1EvtBlkHi - RW – 8 bits - [PM_Reg: 21h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPm1EvtBlkHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O range base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| AcpiPm1EvtBlkHi register. | | | |

| AcpiPm1CntBlkLo - RW – 8 bits - [PM_Reg: 22h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| AcpiPm1CntBlkLo | 7:1 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management Control block. Bit 1 corresponds to Addr[1] and bit 7 corresponds to Addr[7]. |
| AcpiPm1CntBlkLo register. | | | |

| AcpiPm1CntBlkHi - RW – 8 bits - [PM_Reg: 23h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPm1CntBlkHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |

| AcpiPm1CntBlkHi - RW – 8 bits - [PM_Reg: 23h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPm1CntBlkHi register. | | | |

| AcpiPmTmrBlkLo - RW – 8 bits - [PM_Reg: 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| AcpiPmTmrBlkLo | 7:1 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management Timer block. Bit 1 corresponds to Addr[1] and bit 7 corresponds to Addr[7]. |
| AcpiPmTmrBlkLo register. | | | |

| AcpiPmTmrBlkHi - RW – 8 bits - [PM_Reg: 25h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPmTmrBlkHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| AcpiPmTmrBlkHi register. | | | |

| CpuControlLo - RW – 8 bits - [PM_Reg: 26h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 000b | |
| CpuControlLo | 7:3 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management CPU Control block. Bit 3 corresponds to Addr[3] and bit 7 corresponds to Addr[7]. Addr[2:0] are ignored because this register block is 6 byte long. |
| CpuControlLo register. | | | |

| CpuControlHi - RW – 8 bits - [PM_Reg: 27h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CpuControlHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| CpuControlHi register. | | | |

| AcpiGpe0BlkLo - RW – 8 bits - [PM_Reg: 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 00b | |
| AcpiGpe0BlkLo | 7:2 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management General Purpose Event block. Bit 2 corresponds to Addr[2] and bit 7 corresponds to Addr[7]. Addr[1:0] are ignored because this register block is 4 byte long. |
| AcpiGpe0BlkLo register. | | | |

| AcpiGpe0BlkHi - RW – 8 bits - [PM_Reg: 29h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiGpe0BlkHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| AcpiGpe0BlkHi register. | | | |

| AcpiSmiCmdLo - RW – 8 bits - [PM_Reg: 2Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiSmiCmdLo | 7:0 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI SMI Command block. Bit 0 corresponds to Addr[0] and bit 7 corresponds to Addr[7]. The address is required to be DWORD-aligned (Bit[1:0]=00b) |
| AcpiSmiCmdLo register. | | | |

| AcpiSmiCmdHi - RW – 8 bits - [PM_Reg: 2Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiSmiCmdHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| AcpiSmiCmdHi register. | | | |

| AcpiPmaCntBlkLo - RW – 8 bits - [PM_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPmaCntBlkLo | 7:0 | 00h | These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management additional control block. Bit 0 corresponds to Addr[0] and bit 7 corresponds to Addr[7]. |
| AcpiPmaCntBlkLo register. | | | |

| AcpiPmaCntBlkHi - RW – 8 bits - [PM_Reg: 2Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiPmaCntBlkHi | 7:0 | 00h | These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15]. |
| AcpiPmaCntBlkHi register. | | | |

| GEvtConfig0 – RW – 8 bits - [PM_Reg: 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEvtConfig0 | 7:0 | 00h | GEVENT Configuration. These 8 bits are used for configuring general purpose events 0-3. Two bits for each event pin. Bit[1:0] for GEVENT[0], bit[3:2] for GEVENT[1] and so on<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GEVENT to generate SMI#<br>10 GEVENT to generate SMI# followed by SCI<br>11 GEVENT to generate IRQ13 |
| GEvtConfig0 register. | | | |

| GEvtConfig1 – RW – 8 bits - [PM_Reg: 31h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEvtConfig1 | 7:0 | 00h | GEVENT Configuration. These 8 bits are used for configuring General Purpose Events 4-7. Two bits for each event pin. Bit[1:0] for GEVENT[4], bit[3:2] for GEVENT[5] and so on<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GEVENT to generate SMI#<br>10 GEVENT to generate SMI# followed by SCI<br>11 GEVENT to generate IRQ13 |
| GEvtConfig1 register. | | | |

| GPMConfig0 – RW – 8 bits - [PM_Reg: 32h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0Config | 1:0 | 00b | These two bits configure ExtEvent0<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 ExtEvent0 to generate SMI#<br>10 ExtEvent0 to generate SMI# followed by SCI<br>11 ExtEvent0 to generate IRQ13 |
| ExtEvent1Config | 3:2 | 00b | These two bits configure ExtEvent1<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 ExtEvent1 to generate SMI#<br>10 ExtEvent1 to generate SMI# followed by SCI<br>11 ExtEvent1 to generate IRQ13 |
| PCIePmeConfig | 5:4 | 00b | These two bits configure PCIePme<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 PCIePme to generate SMI#<br>10 PCIePme to generate SMI# followed by SCI<br>11 PCIePme to generate IRQ13 |
| Gpm0Config | 7:6 | 00b | These two bits configure GPM0<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GPM0 to generate SMI#<br>10 GPM0 to generate SMI# followed by SCI<br>11 GPM0 to generate IRQ13 |

| GPMConfig1- RW – 8 bits - [PM_Reg: 33h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpm1Config | 1:0 | 00b | These two bits configure GPM1<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GPm1 to generate SMI#<br>10 Gpm1 to generate SMI# followed by SCI<br>11 GPm1 to generate IRQ13 |
| Gpm2Config | 3:2 | 00b | These two bits configure GPM2<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpm2 to generate SMI#<br>10 Gpm2 to generate SMI# followed by SCI<br>11 GPm2 to generate IRQ13 |
| Gpm3Config | 5:4 | 00b | These two bits configure GPM3<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GPM3 to generate SMI#<br>10 GPM3 to generate SMI# followed by SCI<br>11 GPM3 to generate IRQ13 |
| Gpm8Config | 7:6 | 00b | These two bits configure GPM8<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GPM8 to generate SMI#<br>10 GPM8 to generate SMI# followed by SCI<br>11 GPM8 to generate IRQ13 |

| GPMConfig2- RW – 8 bits - [PM_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpio0Config | 1:0 | 00b | These two bits configure Gpio0 or WAKE#<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpio0 to generate SMI#<br>10 Gpio0 to generate SMI# followed by SCI<br>11 Gpio0 to generate IRQ13 |
| Gpm4Config | 3:2 | 00b | These two bits configure GPM4<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpm4 to generate SMI#<br>10 Gpm4 to generate SMI# followed by SCI<br>11 Gpm4 to generate IRQ13 |
| Gpm5Config | 5:4 | 00b | These two bits configure GPM5<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpm5 to generate SMI#<br>10 Gpm5 to generate SMI# followed by SCI<br>11 Gpm5 to generate IRQ13 |
| AzPmeConfig | 7:6 | 00b | These two bits configure AzPme<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 AzPme to generate SMI#<br>10 AzPme to generate SMI# followed by SCI<br>11 AzPme to generate IRQ13 |

| GPMConfig3 – RW – 8 bits - [PM_Reg: 35h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpm6Config | 1:0 | 00b | These two bits configure GPM6<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpm6 to generate SMI#<br>10 Gpm6 to generate SMI# followed by SCI<br>11 Gpm6 to generate IRQ13 |
| Gpm7Config | 3:2 | 00b | These two bits configure GPM7<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpm7 to generate SMI#<br>10 Gpm7 to generate SMI# followed by SCI<br>11 Gpm7 to generate IRQ13 |
| Gpio2Config | 5:4 | 00b | These two bits configure Gpio2<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Gpio2 to generate SMI#<br>10 Gpio2 to generate SMI# followed by SCI<br>11 Gpio2 to generate IRQ13 |
| SataSciConfig | 7:6 | 00b | These two bits configure SataSci<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 SataSci to generate SMI#<br>10 SataSci to generate SMI# followed by SCI<br>11 SataSci to generate IRQ13 |

| GEvtLevelConfig - RW – 8 bits - [PM_Reg: 36h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEvtLevelConfig | 7:0 | 00h | GEVENT input level configuration<br>1 - Rising edge trigger<br>0 - Falling edge trigger |
| GEvtLevelConfig register. | | | |

| GPMLevelConfig0 - RW – 8 bits - [PM_Reg: 37h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0LevelConfig | 0 | 0b | ExtEvent0 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| ExtEvent1LevelConfig | 1 | 0b | ExtEvent1 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| PCIePmeLevelConfig | 2 | 0b | PCIePme input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm0LevelConfig | 3 | 0b | GPM0 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm1LevelConfig | 4 | 0b | GPM1 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm2LevelConfig | 5 | 0b | GPM2 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm3LevelConfig | 6 | 0b | GPM3 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm8LevelConfig | 7 | 0b | GPM8 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |

| GPMLevelConfig1 - RW – 8 bits - [PM_Reg: 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpio0LevelConfig | 0 | 0b | Gpio[0] or WAKE# input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm4LevelConfig | 1 | 0b | GPM4 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm5LevelConfig | 2 | 0b | GPM5 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| AzPmeLevelConfig | 3 | 0b | AzPme input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm6LevelConfig | 4 | 0b | GPM6 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpm7LevelConfig | 5 | 0b | GPM7 input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |
| Gpio2LevelConfig | 6 | 0b | Gpio[2] input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |

| GPMLevelConfig1 - RW – 8 bits - [PM_Reg: 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SataSciLevelConfig | 7 | 0b | SataSci input level configuration<br>1 – Rising edge trigger<br>0 – Falling edge trigger |

| GEvtStatus - RW – 8 bits - [PM_Reg: 39h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEvtStatus | 7:0 | -- | GEVENT status. Write 1 to clear each bit |
| GEvtStatus register. | | | |

| PMEStatus0 - RW – 8 bits - [PM_Reg: 3Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0Status | 0 | 0b | EXTEVENT0 -> SMI# status; write 1 to clear |
| ExtEvent1Status | 1 | 0b | EXTEVENT1 -> SMI# status; write 1 to clear |
| PCIePmeStatus | 2 | 0b | PCIePme -> SMI# status; write 1 to clear |
| Gpm0Status | 3 | 0b | GPM0 -> SMI# status; write 1 to clear |
| Gpm1Status | 4 | 0b | GPM1 -> SMI# status; write 1 to clear |
| Gpm2Status | 5 | 0b | GPM2 -> SMI# status; write 1 to clear |
| Gpm3Status | 6 | 0b | GPM3 -> SMI# status; write 1 to clear |
| Gpm8Status | 7 | 0b | GPM8 -> SMI# status; write 1 to clear |

| PMEStatus1- RW – 8 bits - [PM_Reg: 3Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpio0Status | 0 | 0b | Gpio0 or WAKE# -> SMI# status; write 1 to clear |
| Gpm4Status | 1 | 0b | GPM4 -> SMI# status; write 1 to clear |
| Gpm5Status | 2 | 0b | GPM5 -> SMI# status; write 1 to clear |
| AzPmeStatus | 3 | 0b | AzPme -> SMI# status; write 1 to clear |
| Gpm6Status | 4 | 0b | GPM6 -> SMI# status; write 1 to clear |
| Gpm7Status | 5 | 0b | GPM7 -> SMI# status; write 1 to clear |
| Gpio2Status | 6 | 0b | Gpio2 -> SMI# status; write 1 to clear |
| SataSciStatus | 7 | 0b | SataSci -> SMI# status; write 1 to clear |

| OthersConfig- RW – 8 bits - [PM_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LegacySMIConfig | 1:0 | 00b | These two bits configure legacy SMI# events<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Legacy SMI EVENT to generate SMI#<br>10 Legacy SMI EVENT to generate SMI# followed by SCI<br>11 Legacy SMI EVENT to generate IRQ13 |
| TALERTConfig | 3:2 | 00b | These two bits configure TALERT pin<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 TALERT to generate SMI#<br>10 TALERT to generate SMI# followed by SCI<br>11 TALERT to generate IRQ13 |
| USBConfig | 5:4 | 00b | These two bits configure internal USB PME<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 UsbPme to generate SMI#<br>10 UsbPme to generate SMI# followed by SCI<br>11 UsbPme to generate IRQ13 |

| OthersConfig- RW – 8 bits - [PM_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Ac97Config | 7:6 | 00b | These two bits configure AC97 PME<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 Ac97 PME to generate SMI#<br>10 Ac97 PME to generate SMI# followed by SCI<br>11 Ac97 PME to generate IRQ13 |

| MorePmeConfig – RW – 8 bits - [PM_Reg: 3Dh] | | | |
|---|---|---|---|
| **Fi/eld Name** | **Bits** | **Default** | **Description** |
| OtherThermConfig | 1:0 | 00b | These two bits configure OtherTherm<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 OtherTherm to generate SMI#<br>10 OtherTherm to generate SMI# followed by SCI<br>11 OtherTherm to generate IRQ13 |
| Gpm9Config | 3:2 | 00b | These two bits configure GPM[9]<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 GPM[9] to generate SMI#<br>10 GPM[9] to generate SMI# followed by SCI<br>11 GPM[9] to generate IRQ13 |
| PCIeHotPlugConfig | 5:4 | 00b | These two bits configure PCIeHotPlug<br>00 ACPI Event (trigger SCIOUT or SMI# depending on SCI_EN bit)<br>01 PCIeHotPlug to generate SMI#<br>10 PCIeHotPlug to generate SMI# followed by SCI<br>11 PCIeHotPlug to generate IRQ13 |
| Reserved | 7:6 | 00b | |

| VRT_T1 - RW – 8 bits - [PM_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VRT_T1 | 7:0 | 01h | The RTC battery is sampled (to conserve power) periodically to checks its state of health. VRT_T1 and VRT_T2 make up the interval of the checking. When VRT_Enable is high, the battery is being sampled. When VRT_enable is low, the battery is not being sampled. This register defines the time of VRT enable being high for RTC battery monitor circuit, in miliseconds. |
| VRT_T1 register. | | | |

| VRT_T2 - RW – 8 bits - [PM_Reg: 3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VRT_T2 | 7:0 | FFh | This register defines the time of VRT enable being low for the RTC battery monitor circuit, in 4 ms increment. |
| VRT_T2 register. | | | |

| AD_Pull_UpB - RW – 8 bits - [PM_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AD_Pull_UpB | 7:0 | 00h | This register controls integrated pull-up for AD[31:24] respectively.<br>0: Enable<br>1: Disable |
| AD_Pull_UpB register. | | | |

| AD23_Pull_UpB - RW – 8 bits - [PM_Reg: 41h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AD23_Pull_UpB | 0 | 0b | This bit controls integrated pull-up for AD[23].<br>0: Enable<br>1: Disable |
| Reserved | 7:1 | 0000_000b | |
| AD23_Pull_UpB register. | | | |

| MiscPmControl - RW – 8 bits - [PM_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CPU_IO_PullDownDrvStrength | 0 | 0b | When set, the integrated pull-down drive strength of all CPU IOs are increased by 50% |
| DisableBootFailCpuRst | 1 | 0b | This bit can be used to disable the boot fail timer to generate CPURST#/LDTRST# |
| C2ToC3Enable | 2 | 0b | When set, SB will treat C2 as C3 under K8 mode |
| RstCpuPGEnable | 3 | 0b | If this bit is set to 1, SB toggles CPUPG on every reset |
| Reserved | 6:4 | 000b | |
| SSEnable | 7 | 0b | Enables the spread spectrum PLL |
| MiscPmControl | | | |

| PM_Enable - RW – 8 bits - [PM_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| C3_DPRSLP_EN | 0 | 0b | When set, reading LVL3 register will cause “deeper” sleep for Intel® mobile P4. This is equivalent to LVL4 read. |
| CPU_STP_EN_S2 | 1 | 0b | When set, CPU_STP# will be asserted in S2 |
| CPU_STP_EN_S3 | 2 | 0b | When set, CPU_STP# will be asserted in S3 |
| CPU_STP_EN_S5 | 3 | 0b | When set, CPU_STP# will be asserted in S5 |
| AGP_STP_EN | 4 | 0b | When set, AGP_STP# message will be enabled. |
| PCI_STP_EN_S2 | 5 | 0b | When set, PCI_STP# will be asserted in S2 |
| PCI_STP_EN_S3 | 6 | 0b | When set, PCI_STP# will be asserted in S3 |
| PCI_STP_EN_S5 | 7 | 0b | When set, PCI_STP# will be asserted in S5 |
| PM_Enable register | | | |

| TPRESET1 - RW – 8 bits - [PM_Reg: 51h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TPRESET1 | 5:0 | 00h | Timing parameter used for C states in P4 system. This register defines the wakeup latency in 8µs increment with 8us uncertainty. |
| BmStsRdMask<br>(available after A13) | 6 | 0b | Setting this bit to 1 will keep BM_STS read as 0, unless the BM_STS is caused by USB OHCI and UsbOhciBmStsRdEn is set to 1. |
| StpClkHoldSel<br>(available after A12) | 7 | 0b | Use this bit to select STPCLK# hold time with respect to SLP#<br>0: 200ns<br>1: 3.9us |
| TPRESET1 register | | | |

| TPRESET2 - RW – 8 bits - [PM_Reg: 52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TPRESET2 | 5:0 | 00h | Timing parameter used for S* -> S0 state transitions. This register determines the CPU_STP# deassertion delay in 8µs increment with 8us uncertainty. |
| Reserved | 6 | 0b | |

| TPRESET2 - RW – 8 bits - [PM_Reg: 52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PopUpReqHoldEn (Applicable to ASIC revision A21 and above) | 7 | 0b | Setting this bit to 1 will cause the pop-up request from the NB, or from inside of the SB, to be captured and held until the minimum LDTSTP# assertion time has expired. |
| TPRESET2 register | | | |

| TESTENABLE - RW – 8 bits - [PM_Reg: 53h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| T32_64 | 0 | 0b | Timing parameter used for C3 state in P4 system.<br>If set, the time from the reception of STOP_GNT# to SLP# will be ~128 A-Link clocks. When it is cleared, the timing will be ~64 A-Link clocks. Timing for CPU_STP#/PCI_STP# is controlled similarly, except the value is 192 or 96 A-Link clocks respectively when the bit is set or cleared. |
| Reserved | 2:1 | 00b | |
| SMI_Disable | 3 | 0b | When set, SMI# generation will be disabled |
| Sel_wakeclk | 4 | 0b | Test feature: set 1 to use OSC as the wake up clock; otherwise, use RTC CLK as the wake up clock |
| Reserved | 7:5 | 000b | |
| TESTENABLE register | | | |

| PWRBTTN_CLR - RW – 8 bits - [PM_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TFATAL_CLR | 0 | 0b | Write 1 to clear the TFATAL status bit |
| PWRBTTN_CLR | 1 | 0b | Write 1 to clear the Power Button status bit |
| SHUTDOWN_CLR | 2 | 0b | Write 1 to clear the SHUTDOWN#/GPIO5/SMARTVOLT2 status bit |
| Reserved | 7:3 | 0000_0b | |
| PWRBTTN_CLR register | | | |

| SoftPciRst – RW – 8 bits – [PM_Reg:55h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SoftPciRstEn | 0 | 1b | This bit enables both the soft PCIRST and the THRMTRIP function. |
| Gate_HpetIrq | 1 | 0b | Set to 1 to let HPET enable bit to control IRQ output |
| UserResetEnable | 2 | 1b | When set, GPM7 becomes user reset pin. |
| PCIeNative (Applicable to ASIC revision A21 and above) | 3 | 0b | Setting this bit to 1 will cause PCIeHotPlug PCIePme and WakeAsGevent status to not be set by corresponding events as required by PCIe native mode. |
| PCIeWakeMask (Applicable to ASIC revision A21 and above) | 4 | 0b | Setting this bit will cause PCIEXP_WAKE_STS and PCIEXP_WAKE_DIS to not be visible as required by WinXP. |
| PCIeWakeNoSci (Applicable to ASIC revision A21 and above) | 5 | 0b | Setting this bit will cause PCIEXP_WAKE_STS to not generate SCI. |
| SoftPciRst | 6 | 0b | Setting bit 6 will cause a PCIRST. |
| HideHpetBar | 7 | 0b | Set to1 to make Bar1 in SM configuration space invisible (can Not write and always read 0) |
| SoftPciRst register | | | |

| Reserved – 8 bits - [PM_Reg: 56h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |
| Reserved | | | |

| Reserved – 8 bits - [PM_Reg: 57h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |
| Reserved | | | |

| Reserved – 8 bits - [PM_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |
| Reserved | | | |

| MiscOption - RW – 8 bits - [PM_Reg: 59h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MaskAc97[1:0] | 1:0 | 00b | Setting the bit will mask out the ac97 PCI configuration space (function 5, 6) |
| SataClkSelect | 2 | 0b | 0-external clock, 1-shared with PCIE 100MHz clock |
| AzEnable | 3 | 1b | 1 – Enable HD AUDIO module<br>0 – Disable HD AUDIO module |
| AzSnoopEnable | 4 | 0b | When set, HD AUDIO data transfer will not cause the BM_STS bit to be set and to wake up the CPU from C3 state. This is used when previous version of SB did not have C state pop-up/down function. With the new pop-up/down function in SB600, this function is not necessary and should not be enabled if the C-state pop-up/down is enabled. |
| Gpio5ShutdownEn | 5 | 0b | Shutdown system if seeing a negative edge on Gpio5 |
| Reserved | 7:6 | 00b | |
| MiscOption register | | | |

| SmiSciSts0 - RW – 8 bits - [PM_Reg: 5Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GeventStatus | 7:0 | 00h | These bits indicate the SMI# status of the eight general purpose event signals if they are configured to generate SMI# followed by SCI |
| SmiSciSts0 register | | | |

| SmiSciSts1 - RW – 8 bits - [PM_Reg: 5Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LEventStatus | 0 | 0b | This bit indicates the SMI# status of the legacy power management logic if it is configured to generate SMI# followed by SCI |
| TwarnStatus | 1 | 0b | This bit indicates the SMI# status of the Temperature Caution input if it is configured to generate SMI# followed by SCI |
| Reserved | 2 | 0b | |
| USBStatus | 3 | 0b | This bit indicates the SMI# status of the PME# from the internal USB controller if it is configured to generate SMI# followed by SCI |
| AC97Status | 4 | 0b | This bit indicates SMI# status of the PME# from the internal ac97 controller if it is configured to generate SMI# followed by SCI |
| OtherThermStatus | 5 | 0b | This bit indicates the SMI# status of OtherTherm from NB, fan, etc. if it is configured to generate SMI# followed by SCI |
| GPM9Status | 6 | 0b | This bit indicates the SMI# status of GPM[9] if it is configured to generate SMI# followed by SCI |
| PCIeHotPlugStatus | 7 | 0b | This bit indicates the SMI# status of PCIeHotPlug if it is configured to generate SMI# followed by SCI |
| SmiSciSts1 register | | | |

| SmiSciSts2 - RW – 8 bits - [PM_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtEvent0Status | 0 | 0b | This bit indicates the SMI# status of ExtEvent0 to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| ExtEvent1Status | 1 | 0b | This bit indicates the SMI# status of ExtEvent1 to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| PCIePmeStatus | 2 | 0b | This bit indicates the SMI# status of the PME# from PCIExpress if it is configured to generate SMI# followed by SCI |
| GPM0Status | 3 | 0b | This bit indicates the SMI# status of GPM[0] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM1Status | 4 | 0b | This bit indicates the SMI# status of GPM[1] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM2Status | 5 | 0b | This bit indicates the SMI# status of GPM[2] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM3Status | 6 | 0b | This bit indicates the SMI# status of GPM[3] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM8Status | 7 | 0b | This bit indicates the SMI# status of GPM[8] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| SmiSciSts2 register | | | |

| SmiSciSts3 - RW – 8 bits - [PM_Reg: 5Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpio0Status | 0 | 0b | This bit indicates the SMI# status of GPIO0 (or WAKE#/GEVENT8 pin if PM IO Reg 84h bit1 =1) to SCI/wakeup if it is configured to generate SMI# followed by SCI |
| GPM4Status | 1 | 0b | This bit indicates the SMI# status of GPM[4] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM5Status | 2 | 0b | This bit indicates the SMI# status of GPM[5] to SCI/Wakeup |
| AzaliaStatus | 3 | 0b | This bit indicates the SMI# status from the internal HD Audio controller if it is configured to generate SMI# followed by SCI |
| GPM6Status | 4 | 0b | This bit indicates the SMI# status of GPM[6] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| GPM7Status | 5 | 0b | This bit indicates the SMI# status of GPM[7] to SCI/Wakeup if it is configured to generate SMI# followed by SCI |
| Gpio2Status | 6 | 0b | This bit indicates the SMI# status of GPIO2 to SCI/wakeup if it is configured to generate SMI# followed by SCI |
| SataSciStatus | 7 | 0b | This bit indicates the SMI# status of SataSci to SCI/wakeup |
| SmiSciSts3 register | | | |

| MwaitEnable - RW – 8 bits - [PM_Reg: 5Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mwait_any_smi_en | 0 | 0b | SMI# is generated when any CPU is in mwait state if this bit is set to 1 |
| Mwait_2cpu_smi_en | 1 | 0b | For 2 CPU system (dual core, non HT) SMI# is generated when both CPUs are in mwait state if this bit is set to 1 |
| Mwait_4cpu_smi_en | 2 | 0b | For 4 CPU system (dual core, HT) SMI# is generated when all 4 CPUs are in mwait state if this bit is set to 1 |
| Mwait_2cpu_C23_en | 3 | 0b | For 2 CPU system (dual core, non HT) C2 or C3 is generated when both CPUs are in mwait state if this bit is set to 1. C2 or C3 is determined by ARB_DIS = 0 or 1. |
| Mwait_4cpu_C23_en | 4 | 0b | For 4 CPU system (dual core, HT) C2 or C3 is generated when all 4 CPUs are in mwait state if this bit is set to 1. C2 or C3 is determined by ARB_DIS = 0 or 1. |
| Reserved | 7:5 | 000b | |

| MwaitEnable - RW – 8 bits - [PM_Reg: 5Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MwaitEnable register<br>This register is used only in the P4 system. | | | |

| MwaitSmiSts - RW – 8 bits - [PM_Reg: 5Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mwait_any_smi_sts | 0 | 0b | This bit indicates that SMI# is generated when any CPU is in mwait state |
| Mwait_2cpu_smi_sts | 1 | 0b | This bit indicates that for 2 CPU system (dual core, non HT) SMI# is generated when both CPUs are in mwait state |
| Mwait_4cpu_smi_sts | 2 | 0b | This bit indicates that for 4 CPU system (dual core, HT) SMI# is generated when all 4 CPUs are in mwait state |
| Reserved | 7:3 | 00h | |
| MwaitSmiSts register<br>This register is used only in the P4 system. | | | |

| Options_0 - RW – 8 bits - [PM_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TAlertFanEn | 0 | 0b | Set this bit to put all fans to full speed if TALERT# is asserted |
| ProcHotFanEn | 1 | 0b | Set this bit to put all fans to full speed if PROCHOT# is asserted |
| Fan1En | 2 | 0b | Setting this bit will configure GPIO48 pin to be FAN1 output |
| Fan2En | 3 | 0b | Setting this bit will configure GPIO49 pin to be FAN2 output |
| Reserved | 4 | 0b | |
| SpkrEn | 5 | 0b | Setting this bit will configure GPIO2 to be speaker output |
| Fan0En | 6 | 0b | Setting this bit will configure GPIO3 to be FAN0 output |
| Reserved | 7 | 0b | |
| Options_0 register | | | |

| Options_1 - RW – 8 bits - [PM_Reg: 61h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| IsAmd | 1 | 0b | Set to enable NB/SB handshake during IOAPIC interrupt for AMD K6 or K7 class; Clear for other CPU. |
| PCI_Active_enable | 2 | 0b | BIOS should set this bit in order to monitor BM_STS pin from NB (the pin is called BMREQ# on SB) and bus mastering from the SB itself. |
| UseCpuRst | 3 | 1b | If this bit is not set, then system reset will cause INIT# instead of CPURST#. |
| ProcHotStsEn | 4 | 0b | Set to enable PROCHOT# to generate TwarnStatus and thermal throttle |
| Reserved | 5 | 0b | |
| UsbPmeEnable | 6 | 0b | USB PME enable |
| Reserved | 7 | 0b | |
| Options_1 register | | | |

| Shadow_SCI- R – 8 bits - [PM_Reg: 62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Shadow_SCI | 0 | 0b | SCI output |
| Reserved | 7:1 | 00h | |
| Shadow_SCI register | | | |

| SwitchVoltageTime - RW – 8 bits - [PM_Reg: 63h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SwitchVoltageTime | 5:0 | 05h | Programmable value (in 2us increment with 2us uncertainty) |
| Reserved | 7:6 | 00b | |
| SwitchVoltageTime register | | | |

| SwitchGHI_Time - RW – 8 bits - [PM_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SwitchGhiTime | 5:0 | 02h | Programmable value (in 2us increment with 2us uncertainty) |
| Reserved | 7:6 | 00b | |
| SwitchGHITime register | | | |

| UsbPMControl- RW – 8 bits - [PM_Reg: 65h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| UsbPhyS5PwrDwnEnable | 0 | 0b | Set to 1 to enable S4/S5 USB Phy power down support and to disable S4 USB wakeup support.<br>The bit has to be clear to 0 (default) to support S4 USB wakeup. |
| Reserved | 1 | 0b | |
| UsbResumeEnable | 2 | 1b | Set to 1 to enable S3 wakeup on USB device resume |
| Reserved | 3 | 0b | |
| UsbResetByPciRstEnable | 4 | 1b | Set to 1 to reset USB on the software (such as IO-64 or IO-CF9 cycles) generated PCIRST#. |
| UsbS5ResetEnable | 5 | 1b | Set to 1 to enable USB reset on S4/S5 resume detection |
| Reserved | 6 | 0b | |
| SpecialFunc | 7 | 0b | If set to 1, S* -> S0 state transitions will use 1ms clock for timing sequence; otherwise, 8μs clock will be used.<br>For K8 system, this bit must be cleared to use 8μs clock. |
| UsbPMControl register | | | |

| MiscEnable66 - RW – 8 bits - [PM_Reg: 66h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| UsbBusyBreakEn (available after A13) | 0 | 0b | Setting this bit to 1 will cause C3/4 wakeup when USB OHCI or EHCI DMA is active. |
| UsbOhciCstateMask (available after A13) | 1 | 0b | Setting this bit to 1 will prevent the system from entering C state when USB OHCI DMA is active. |
| Reserved | 4:2 | 000b | |
| K8KbRstEn | 5 | 0b | KB_RST# control for K8 system<br>0: Generate INIT#<br>1: Generate PCIRST# |
| UsbBusyBmStsEn | 6 | 0b | Setting this bit to 1 will cause C3/4 pop-up when USB OHCI or EHCI DMA is active. |
| UsbOhciBmStsRdEn (available after A13) | 7 | 0b | Setting this bit to 1 will keep BM_STS read as 1 regardless of BmStsRdMask when USB OHCI DMA is active. |
| MiscEnable66 register | | | |

| MiscEnable67 – RW – 8 bits – [PM_Reg:67h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CPU_STP_En | 0 | 0b | CPU_STP# enable for C3 state in P4 system; no effect in K8 systems, for which CPU_STP# is always deasserted. |
| Slp_En | 1 | 0b | SLP# enable in C states |
| CC_En | 2 | 0b | C State enable; must be set in order to exercise C state |
| Reserved | 3 | | |
| BypassPwrGoodEn | 4 | 0b | If asserted, Southbridge will not wait for deassertion of PWRGOOD to monitor for wakeup event |

| MiscEnable67 – RW – 8 bits – [PM_Reg:67h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TempPolarity | 6:5 | 00b | Temperature polarity control for THRMTRIP and TALERT respectively. 0: active low; 1; active high |
| DlySlpEn | 7 | 0b | Set to 1 to delay recognition of STPGNT# until there is no pending read in AB |
| MiscEnable67 register | | | |

| MiscEnable68 – RW – 8 bits – [PM_Reg:68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NicePBE | 0 | 0b | When set, the CPU 's assertion of PBE# during C state will cause SB to break out from C state gracefully; i.e., it will finish the C state entry sequence first before it will revert back to C0. If this bit is not set, it will break out from C state abruptly. It is recommended to set this bit. |
| MaskCState | 1 | 0b | (Use this only for the P4 system. Leave it 0 for the K8 system). If this bit is set, the pending break event will prevent the SB from entering C state. The LVL read will be completed with no effect. |
| MaskApicEn | 2 | 0b | If set, the APIC interrupt will be deferred until the first ACPI access when the system resumes from S state. For K8 CPU additionally A20M# IGNNE# INTR NMI INIT# will be deferred the same way, but SMI# will not be deferred. |
| THRMTRIP_Enable | 3 | 1b | When set, GEVENT2 becomes THRMTRIP function. When THRMTRIP pin is low and PM 55h, bit 0 is set, hardware will switch the system to S5 automatically. |
| MaskRtcClkOut | 4 | 0b | If set, RtcClkOut will stop toggling. |
| LLB_Enable | 5 | 0b | If set, LLB function is enabled, system won't wakeup from ACPI S state until LLB# is de-asserted. |
| AcpiThrotPeriod | 7:6 | 00b | Selects the software clock throttling period<br>00: 15μs<br>01: 30μs<br>10: 244μs<br>11: Reserved |
| MiscEnable68 register | | | |

| WatchDogTimerControl – RW – 8 bits – [PM_Reg:69h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogTimerDisable | 0 | 1b | When set, watchdog timer will be disabled |
| Reserved | 7:1 | 00h | |
| WatchDogTimerControl register | | | |

| WatchDogTimerBase0 – RW – 8 bits – [PM_Reg:6Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 000b | |
| WatchDogTimerBase0 | 7:3 | 00000b | WatchDogTimer Base address [7:3] |
| WatchDogTimerBase0 register | | | |

| WatchDogTimerBase1 – RW – 8 bits – [PM_Reg:6Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogTimerBase1 | 7:0 | 00h | WatchDogTimer Base address [15:8] |
| WatchDogTimerBase1 register | | | |

| WatchDogTimerBase2 – RW – 8 bits – [PM_Reg:6Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogTimerBase2 | 7:0 | 00h | WatchDogTimer Base address [23:16] |
| WatchDogTimerBase2 register | | | |

| WatchDogTimerBase3 – RW – 8 bits – [PM_Reg:6Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogTimerBase3 | 7:0 | 00h | WatchDogTimer Base address [31:24] |
| WatchDogTimerBase3 register | | | |

| S_LdtStartTime – RW – 8 bits – [PM_Reg:70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S_LdtStartTime | 7:0 | 00h | This register defines the delay between SUS_STAT# assertion and LDTSTP# assertion when the K8 system enters ACPI S states, in 1us increment, with 1us uncertainty. |
| S_LdtStartTime register | | | |

| EnhanceOption – RW – 8 bits – [PM_Reg:71h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| P4C34PopUpEn | 1 | 0b | If enabled, for P4 system C3/4 can pop up to C2 for internal DMA request and back down to C3/4 after A-link bus is idle for number of clocks defined by PopUpEndTime.<br>1 = Enable<br>0 = Disable |
| C2EnhanceEn | 2 | 0b | For both P4 and K8 system, in C2 state NB can toggle SLP#/LDTSTP#. When entering C2 state, SB sends out STPCLK# assertion message. NB takes control of SLP#/LDTSTP#. When exiting C2 state, SB sends out STPCLK# de-assertion message. NB de-assert SLP#/LDTSTP# if needed. If this bit is enabled, SB will wait for NB to send the same message back then de-assert STPCLK# signal for P4 system or send another STPCLK# de-assertion message for K8 system. If this bit is disabled, SB will not wait for NB to send the message back.<br>1 = Enable<br>0 = Disable |
| Reserved | 3 | 0b | |
| VidFidExtraDelayEn | 4 | 0b | If enabled, extra duration of LDTSTP# assertion as specified by VidFidExtraDelaySelect will be added to the VID/FID change sequence.<br>1 = Enable<br>0 = Disable |
| VidFidExtraDelaySelect | 7:5 | 000b | 3’b000: 0ns<br>3'b001: 140ns<br>3'b010: 210ns<br>3'b011: 280ns<br>3'b110: 350ns<br>3'b111: 420ns<br>3'b100: 490ns<br>3'b101: 560ns |
| EnhanceOption register | | | |

| C4Control – RW – 8 bits – [PM_Reg:72h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DPRSLPVR_delay | 6:0 | 000_0000b | This defines the delay between CPU_STP# de-assertion and DPRSLPVR de-assertion in C4 state, in 2us increment with 2us uncertainty. |
| C4ChkVgate | 7 | 0b | 1: check VGATE=1 before DPRSLPVR de-assertion<br>0: use DPRSLPVR_delay for DPRSLPVR de-assertion |
| C4Control register for the P4 system. | | | |

| PopUpEndTime – RW – 8 bits – [PM_Reg:73h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PopUpEndTime | 7:0 | 80h | This register applies to both P4 and K8 system. During C3/4 pop-up, SB monitors DMA traffic. If there has been no traffic for PopUpEndTime, SB will bring system back to C3/4.<br>For P4 system, the time is counted by A-link clock.<br>For K8 system, the time is counted by OSC clock. |
| PopUpEndTime register | | | |

| PwrFailShadow – RW – 8 bits – [PM_Reg:74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PwrFailShadow | 3:0 | 0h | Writing to these four bits will set the value onto bits [7:4].<br>Software should always set bit 2 = 1 |
| PwrFailOption | 5:4 | 00b | These two bits will determine how system should resume after a power failure.<br>00: Use default: Wait for wake event to power up<br>01: Always on: always power on after power resumes<br>10: Always off: always power off after power resumes<br>11: Use previous: resume to same setting when power fails. |
| PowerState | 6 | 0b | Power state indicator: 1 = on; 0 = off |
| ForcePwrOn | 7 | 0b | If set and RTC AIE =1, will force power on after power resumes regardless of Bit[5:4] setting; if cleared and RTC AIE = 1, will wakeup when RTC alarm fires after a power failure/resume. |
| PwrFailShadow register<br>Note: For ASIC revisions A11, A12, and A13 this register needs to be written twice for bits [7:4] to be updated correctly. There is no such restriction with other revisions. | | | |

| Tpreset1b – RW – 6 bits – [PM_Reg:75h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Tpreset1b | 5:0 | 05h | Timing parameter used for S* -> S0 state transition.<br>This determines the delay between CPU_STP# de-assertion and SUS_STAT# de-assertion, in 8us increment with 8us uncertainty. |
| DelayRomRstEn | 6 | 1b | 1: 30ms ahead of deassertion of PciRst#<br>0: ROM_RST functions the same as PciRst# |
| DelayLanRstEn | 7 | 1b | 1: 20ms ahead of deassertion of PciRst#<br>0: Lan_RST functions the same as PciRst# |

| S0S3ToS5Enable0 – RW – 8 bits – [PM_Reg:76h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S0S3ToS5Enable0 | 7:0 | FFh | This register determines which wakeup signals can be passed onto the S5 region. For S4/S5 wake up, this bit must be set in before the corresponding pin can be used<br>Bit [7:0]: GEVENT#[7:0]<br>1: Enable<br>0: Disable |

| S0S3ToS5Enable0 – RW – 8 bits – [PM_Reg:76h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S0S3ToS5Enable0 register | | | |

| S0S3ToS5Enable1 – RW – 8 bits – [PM_Reg:77h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S0S3ToS5Enable1 | 7:0 | FFh | This register determines which wakeup signals can be passed onto the S5 region. For S4/S5 wake up, this bit must be set in before the corresponding pin can be used<br>Bit [7:0]: GPM#[7:0]<br>1: Enable<br>0: Disable |
| S0S3ToS5Enable1 register | | | |

| S0S3ToS5Enable2 – RW – 8 bits – [PM_Reg:78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S0S3ToS5Enable2 | 7:0 | FFh | This register determines which wakeup signals can be passed onto the S5 region. For S4/S5 wake up, this bit must be set in before the corresponding pin can be used<br>Bit [0]: Spare<br>Bit [1]: Internal USB<br>Bit [2]: EXTEVENT#[0]<br>Bit [3]: EXTEVENT#[1]<br>Bit [4]: External IRQ8<br>Bit [5]: GPIO[0]<br>Bit [6]: GPIO[1]<br>Bit [7]: GPIO[2]<br>1: Enable<br>0: Disable |
| S0S3ToS5Enable2 register | | | |

| S0S3ToS5Enable3 – RW – 8 bits – [PM_Reg:79h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S0S3ToS5Enable3 | 7:0 | FFh | This register determines which wakeup signals can be passed onto the S5 region. For S4/S5 wake up, this bit must be set in before the corresponding pin can be used<br>Bit [0]: GPIO[3]<br>Bit [1]: Spare<br>Bit [2]: SATA_SCI<br>Bit [3]: Spare<br>Bit [4]: Spare<br>Bit [5]: Spare<br>Bit [6]: S5ResetOverride<br>Bit [7]: Enable bit to pass PCI config gevent1_en* bits<br>1: Enable<br>0: Disable |
| S0S3ToS5Enable3 register | | | |

| NoStatusControl0 – RW – 8 bits – [PM_Reg:7Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NoStatusControl0 | 7:0 | 00h | For GEVENT#[7:0] configured as PME# (wakeup function), additional setting of these bits will still allow GEVENT# pins to wakeup the system but they will not set the status bit in the ACPI GEVENT status register.<br>Bits [7:0] control GEVENT#[7:0] respectively<br>0 – Allow status<br>1 – Not-allow status |
| NoStatusControl0 register | | | |

| NoStatusControl1 – RW – 8 bits – [PM_Reg:7Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NoStatusControl1 | 7:0 | 00h | This register should be set the same as 7A |
| NoStatusControl1 register | | | |

| MiscEnable7C – RW – 8 bits – [PM_Reg:7Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BreakEnable | 0 | 0b | Set to 1 to allow wakeup from C3 without asserting LDTSTP# in K8 system |
| Reserved | 1 | 0b | |
| BlinkControl | 3:2 | 00b | Blinking interval select<br>00: Always off<br>01: 1sec on, 3sec off, repeating<br>10: 2sec on, 2sec off, repeating<br>11: Always on |
| AltCenturyEnable | 4 | 0b | Set to 1 to enable RTC AltCentury register |
| SusStatOption | 5 | 0b | SUS_STAT# timing option enable |
| WaitStpGntEnB | 6 | 0b | 0: Wait for STPGNT# in ACPI S state;<br>1: No wait |
| Reserved | 7 | 0b | |
| MiscEnable7C register | | | |

| DprSlpVrMinTime – RW – 8 bits – [PM_Reg:7Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DprSlpVrMinTime | 3:0 | 2h | This defines the minimum assertion time of DPRSLPVR, in 2us increment with 2us uncertainty. |
| Reserved | 7:4 | 0h | |
| DprSlpVrMinTime register | | | |

| SMAF0 – RW – 8 bits – [PM_Reg:80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S45SMAF | 2:0 | 110b | System management action field for S4/5 STPCLK message |
| Reserved | 3 | 0b | |
| C2SMAF | 6:4 | 0h | System management action field for C2 STPCLK message |
| Reserved | 7 | 0b | |
| SMAF0 register | | | |

| SMAF1 – RW – 8 bits – [PM_Reg:81h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| C3SMAF | 2:0 | 001b | System management action field for C3 STPCLK message |
| Reserved | 3 | 0b | |
| VFSMAF | 6:4 | 010b | System management action field for VFID STPCLK message |
| Reserved | 7 | 0b | |

| SMAF1 – RW – 8 bits – [PM_Reg:81h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMAF1 register | | | |

| SMAF2 – RW – 8 bits – [PM_Reg:82h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| S1SMAF | 2:0 | 011b | System management action field for S1 STPCLK message |
| Reserved | 3 | 0b | |
| S3SMAF | 6:4 | 100b | System management action field for S3 STPCLK message |
| Reserved | 7 | 0b | |
| SMAF2 register | | | |

| SMAF3 – RW – 8 bits – [PM_Reg:83h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| NSSMAF | 2:0 | 101b | System management action field for Normal Throttling STPCLK message |
| Reserved | 3 | 0b | |
| TTSMAF | 6:4 | 101b | System management action field for Thermal Throttling STPCLK message |
| Reserved | 7 | 0b | |
| SMAF3 register | | | |

| WakePinCntl – RW – 8 bits – [PM_Reg:84h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WakePinEnable | 0 | 1b | Set to 1 to enable wakeup from WAKE#/GEVENT#[8] pin |
| WakePinAsGEvent | 1 | 0b | Set to 1 to use WAKE#/GEVENT#[8] pin as GEvent, replacing GPIO0 |
| WakeOut | 2 | 0b | Output data for wake pin |
| WakeOutEnB | 3 | 1b | Output enable for wake pin (active low) |
| WakePinStatus | 4 | 0b | Input status for wake pin |
| Reserved | 7:5 | 000b | |
| WakePinCntl register | | | |

| CF9Rst – RW – 8 bits – [PM_Reg:85h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| SysRst | 1 | 0b | 0: Send INIT HT message<br>1: Reset as specified by bit3 |
| RstCmd | 2 | 0b | Write with 1 to generate reset as specified by bit[3,1]<br>Write only, always read as 0 |
| FullRst | 3 | 0b | 0: Assert reset signals only<br>1: Place system in S5 state for 3 to 5 seconds |
| Reserved | 7:4 | 0h | |
| CF9Rst register | | | |

| ThermThrotCntl – RW – 8 bits – [PM_Reg:86h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Therm2SecDelay | 0 | 0b | Enable 2 second delay for thermal clock throttle |

| ThermThrotCntl – RW – 8 bits – [PM_Reg:86h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ThrottleControl | 4:1 | 0000b | Bit[4] Enable thermal clock throttle<br>Bit[3:1] Throttle interval for STPCLK# de-assertion<br>000b: 50%<br>001b: 12.5%<br>010b: 25%<br>011b: 37.5%<br>100b: 50%<br>101b: 62.5%<br>110b: 75%<br>111b: 87.5% |
| ThermThrotPeriod | 5 | 0b | Selects the thermal clock throttle period<br>0: 30us<br>1: 244us |
| Reserved | 6 | 0b | |
| PllRstEn<br>(available after A13) | 7 | 0b | Resets PLL whenever the system gets reset. |
| ThermThrotCntl register | | | |

| LdtStpCmd – RW – 8 bits – [PM_Reg:87h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LSCmd | 0 | 0b | Write 1 to generate VID/FID change sequence with C3 STPCLK# message |
| Reserved | 7:1 | 00h | |
| LdtStpCmd register | | | |

| LdtStartTime – RW – 8 bits – [PM_Reg:88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LdtStartTime | 7:0 | 00h | LDTSTP# assertion delay in 1us increment |
| LdtStartTime register for ACPI C state in the K8 system. | | | |

| AgpStartTime – RW – 8 bits – [PM_Reg:89h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AgpStartTime | 7:0 | 00h | AGP_STP# assertion delay in 1us increment |
| AgpStartTime register for ACPI C state in the K8 system. | | | |

| LdtAgpTimeCntl – RW – 8 bits – [PM_Reg:8Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LdtEndTime | 1:0 | 00b | LDTSTP# de-assertion delay select<br>00: 0μs<br>01: 1μs<br>10: 32μs<br>11: 64μs |
| Reserved | 3:2 | 00b | |
| VfidTime | 6:4 | 000b | VID/FID LDTSTP# duration select<br>000: 1μs<br>001: 2μs<br>010: 4μs<br>011: 8μs<br>100: 16μs<br>101: 32μs<br>110: 64μs<br>111: 128μs |
| StutterMode | 7 | 0b | Set to 1 to enable stutter mode |

| LdtAgpTimeCntl – RW – 8 bits – [PM_Reg:8Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| LdtAgpTimeCntl register for ACPI C state in the K8 system. | | | |

| StutterTime – RW – 8 bits – [PM_Reg:8Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| StutterTime | 7:0 | 00h | LDTSTP# duration in 1us increment |
| StutterTime register for ACPI C state in the K8 system. | | | |

| StpClkDlyTime – RW – 8 bits – [PM_Reg:8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| StpClkDlyTime | 7:0 | 00h | STPCLK# deassertion delay in number of OSC clocks for S1 resume in K8 system. |
| StpClkDlyTime register for ACPI S state | | | |

| MiscEnable8D – RW – 8 bits – [PM_Reg:8Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PmeMsgEn | 0 | 0b | Set to 1 to enable PmeTurnOff/PmeMsgAck handshake |
| SlpS2En | 1 | 0b | Set to 1 to enable GPM9 as SLP_S2 output |
| AzRstEn | 2 | 0b | Set to 1 to enable GPM8 as AZ_RST# output |
| DprStpEn | 3 | 0b | Set to 1 to enable DPRSTP# output |
| Ac97PciClkEnB | 4 | 0b | Clear to 0 to enable gating of the ac97 internal clock |
| LongReset | 5 | 0b | Set to 1 will cause all reset to be 6ms longer |
| EffMaskEn | 6 | 0b | (Set to 1 for the P4 system and leave it 0 for K8 system).<br>On the P4 platform, right after the SB exits from the C state, the LVL read will have no effect until the previous break event has been cleared. Setting the bit will prevent the SB from sending the STPCLK# message to the NB during such time.<br>On the K8 platform the bit has no meaning and must be 0. |
| Reserved | 7 | 0b | |
| AbPmeCntl register | | | |

| FakeAsr– RW – 8 bits – [PM_Reg:8Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FakeAsr | 7:0 | 00h | Fake alternate status |
| FakeAsr register | | | |

| FakeAsrEn– RW – 8 bits – [PM_Reg:8Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FakeAsrEn | 0 | 0b | FakeAsr enable. If set, SB will return a fake status (defined in PMIO_8Eh) when IDE DMA is actively busy and software tries to read the status from the IDE device |
| UseBypassRom | 1 | 0b | When this bit is set, it will override the ROM straps and use bits 3:2 of this register to determine which type of ROM to use. This is for BIOS debugging purpose or for if system desires to have multiple BIOS on board |
| BypassRomSel | 3:2 | 00b | These two bits will override the two ROM strap pins.<br>00 – LPC ROM<br>10 – FWH ROM<br>11 – SPI ROM<br>01 – PCI ROM (not used after A13) |
| BmReqPopUpEn | 4 | 0b | When set (along with C state PopUp enabled), BmReq# input will not cause BM_STS to be set; instead, it will simply act just like AllowLdtStop and cause the C state machine to pop-up to C2. This bit is only valid under K8 CPU configuration. |

| FakeAsrEn– RW – 8 bits – [PM_Reg:8Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MaskNbBmStsSet | 5 | 0b | When set, BmStsSet message from NB will not cause wake up from C state. |
| MemRstDisable | 6 | 0b | When set, the memory reset function at DDR_RST# pin will be disabled and can be used as GPM5# |
| DisableOtherIrq8 | 7 | 0b | When set, other source of interrupt 8 (serial IRQ, HPET) will not be routed to the SCI. Only RTC interrupt will be routed to the SCI |
| FakeAsrEn register | | | |

| GEVENTOUT – RW – 8 bits – [PM_Reg:90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEVENTOUT | 7:0 | 00h | GEVENT[7:0] output value |
| GEVENTOUT register | | | |

| GEVENTEN – RW – 8 bits – [PM_Reg:91h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEVENTEN | 7:0 | FFh | GEVENT[7:0] output enable<br>0: Enable<br>1: Tristate |
| GEVENTEN register | | | |

| GEVENTIN – RW – 8 bits – [PM_Reg:92h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GEVENTIN | 7:0 | -- | GEVENT[7:0] input status |
| GEVENTIN register | | | |

| GPM98OUT – RW – 8 bits – [PM_Reg:94h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPM98OUT | 1:0 | 00b | GPM[9:8] output value |
| Reserved | 7:2 | | |
| GPM98OUT register | | | |

| GPM98EN – RW – 8 bits – [PM_Reg:95h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPM98EN | 1:0 | 11b | GPM[9:8] output enable<br>0: Enable<br>1: Tristate |
| Reserved | 7:2 | 00h | |
| GPM98EN register | | | |

| GPM98IN – RW – 8 bits – [PM_Reg:96h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GPM98IN | 1:0 | -- | GPM[9:8] input status |
| Reserved | 7:2 | 00h | |
| GPM98IN register | | | |

| K8C1ePort - RW - 16 bits - [PM_Reg: 99:98h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| K8C1ePort | 15:0 | 0000h | This register defines the 16 bit IO address for the K8 C1e support. In AMD K8 dual core system, when both CPUs have entered the C1e state, it will broadcast an IO cycle. BIOS can program CPU with this address for such function. When SB receives this IO cycle, it can automatically sequence to C2 or C3 depending on PMIO9Ah, bits [1:0]. |
| K8C1ePort register | | | |

| EnhanceControl - RW - 8 bits - [PM_Reg: 9Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| K8C1eToC2En | 0 | 0b | If this bit is set, a write to the IO address defined by K8C1ePort will cause SB to automatically sequence to C2 |
| K8C1eToC3En | 1 | 1b | If this bit is set, a write to the IO address defined by K8C1ePort will cause SB to automatically sequence to C3. It is BIOS's responsibility not to set both bits 0 and 1 to 1; only one bit can be set to 1. |
| K8C3PopUpEn | 2 | 0b | If this bit is set, SB can pop up from C3 to C2 in the case of bus mastering (DMA) instead of transitioning to C0. After DMA, it goes back to C3 with minimum delay of LdtStartTime. |
| AutoArbDisEn | 3 | 0b | K8 system –<br>If this bit is set and K8C1eToC3En is also set, a write to the IO address defined by K8C1ePort will cause SB automatically set the ARB_DIS bit.<br>P4 system –<br>If this bit is set, LVL3 read will cause SB automatically set the ARB_DIS bit. |
| auto_bm_rld | 4 | 0b | If this bit is set, BM_STS will cause SB to wakeup from C3/4 even if BM_RLD is not set. |
| auto_clr_bm_sts | 5 | 0b | If this bit is set, BM_STS will be cleared when system enters C3/4. |
| FastReadEnable | 6 | 0b | This bit is to make the read from ADC as fast as possible. |
| HPET_Periodic | 7 | 0b | Set to 1 to enable HPET periodic mode hw support. |
| EnhanceControl register | | | |

| K8C1eReadPort - R - 8 bits - [PM_Reg: 9Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| K8C1eReadPort | 7:0 | 00h | The C1e Write by CPU is stored in this register. The value written by the C1e write can also be read from the defined IO address |
| K8C1eReadPort register | | | |

| MsiSignature - R - 24 bits - [PM_Reg: 9E:9Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MsiSignature | 19:0 | 00000h | This defines the MSI signature SB will monitor. This corresponds to address [39:20]. When SB sees transaction with this address, it will issue a break event to the C state machine if the CPU is in C2/3/4 state |
| Spare | 21:20 | 00b | |
| HPET_Version | 23:22 | 00b | Set to 11b to make HPET revision id to be 01. |
| MsiSignature register | | | |

| AutoArbDisWaitTime - RW - 8 bits - [PM_Reg: 9Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AutoArbDisWaitTime | 3:0 | 0h | This defines the amount of time (in 2us increment) that SB will hold ARB_DIS set after breaking from C3. This is to allow sometime for CPU to resume from C3 before allowing any bus mastering to the memory. This timer has an uncertainty of -2us. This applies to K8 C1e or P4 LVL3 if AutoArbDisEn is set. |
| HPET_DisablePeriodic | 4 | 0h | Set to 1 to make Periodical capability bit appear to be 0. |
| HPET_Load | 5 | 0h | Set to 1 to make HPET timer load the new value in periodical mode |
| ASFRemoteDelay | 6 | 0h | Set to 1 to delay the remote action(reset,power dwon..) |
| RstCstate/9E | 7 | 0h | Set to 1 to make state machine of C state reset by pcirst, otherwise rsmrst |
| AutoArbDisWaitTime register | | | |

| ProgramIo4RangeLo - RW – 8 bits - [PM_Reg: A0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo4Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits [2:0]. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo4RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo4RangeLo register | | | |

| ProgramIo4RangeHi - RW – 8 bits - [PM_Reg: A1h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo4RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo4RangeHi register | | | |

| ProgramIo5RangeLo - RW – 8 bits - [PM_Reg: A2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo5Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits [2:0]. For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo5RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo5RangeLo register | | | |

| ProgramIo5RangeHi - RW – 8 bits - [PM_Reg: A3h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo5RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |

| ProgramIo5RangeHi - RW – 8 bits - [PM_Reg: A3h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo5RangeHi register | | | |

| ProgramIo6RangeLo - RW – 8 bits - [PM_Reg: A4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo6Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits [2:0].<br>For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo6RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo6RangeLo register | | | |

| ProgramIo6RangeHi - RW – 8 bits - [PM_Reg: A5h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo6RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo6RangeHi register | | | |

| ProgramIo7RangeLo - RW – 8 bits - [PM_Reg: A6h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo7Mask | 3:0 | 0h | These four bits are used to mask the least 4 bits of the 16 bit I/O. If bit [3] is set, then bit [3] of the I/O address is not compared. If it is not set, then bit [3] of the monitored address is 0. The same applies for the other three bits [2:0].<br>For example, if x15=80h, x14[7:4]=Ah, and x14[3:0]=3h, then the monitored range is 80A4h : 80A0h (bit 0 and 1 are masked) |
| ProgramIo7RangeLo | 7:4 | 0h | I/O range base address; these bits define the least significant byte of the 16 bit I/O range base address that is programmed to trigger SMI# when the address is accessed. Bit 7 corresponds to Addr[7] and bit 4 to Addr[4]. |
| ProgramIo7RangeLo register | | | |

| ProgramIo7RangeHi - RW – 8 bits - [PM_Reg: A7h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo7RangeHi | 7:0 | 00h | I/O range base address; these bits define the most significant byte of the 16 bit I/O range base address. Bit 7 corresponds to Addr[15] and bit 0 to Addr[8]. |
| ProgramIo7RangeHi register | | | |

| PIO7654Enable - RW – 8 bits - [PM_Reg: A8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo4Enable | 0 | 0b | Enables IO monitoring for ProgramIO4 (defined by index A0, A1).<br>1 = On<br>0 = Off |

| PIO7654Enable - RW – 8 bits - [PM_Reg: A8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo5Enable | 1 | 0b | Enables IO monitoring for ProgramIO5 (defined by index A2, A3).<br>1 = On<br>0 = Off |
| ProgramIo6Enable | 2 | 0b | Enables IO monitoring for ProgramIO6 (defined by index A4, A5).<br>1 = On<br>0 = Off |
| ProgramIo7Enable | 3 | 0b | Enables IO monitoring for ProgramIO7 (defined by index A6, A7).<br>1 = On<br>0 = Off |
| Spare | 7:4 | 0h | Scratch bits |
| PIO7654Enable register | | | |

| PIO7654Status - RW – 8 bits - [PM_Reg: A9h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ProgramIo4Status | 0 | 0b | Programmable IO 4 status bit; write 1'b1 to clear the status bit |
| ProgramIo5Status | 1 | 0b | Programmable IO 5 status bit; write 1'b1 to clear the status bit |
| ProgramIo6Status | 2 | 0b | Programmable IO 6 status bit; write 1'b1 to clear the status bit |
| ProgramIo7Status | 3 | 0b | Programmable IO 7 status bit; write 1'b1 to clear the status bit |
| Spare | 7:4 | 0h | Scratch bits |
| PIO7654Status register | | | |

| PllDebug– R/W – 8 bits - [PM_Reg: B0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PLL_debug_delay (available after A13) | 7:0 | 08h | These bits are for PLL debugging purposes. |
| PllDebug register | | | |

| PllDebug– R/W – 8 bits - [PM_Reg: B1h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PLL_debug_range (available after A13) | 5:0 | 00_1001b | These bits are for PLL debugging purposes. |
| Spare | 7:6 | 00b | Scratch bits |
| PllDebug register | | | |

| AltDebugBusCntrl – R/W – 8 bits - [PM_Reg: B2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Disable IDE block (available after A13) | 0 | 0b | This bit is used to disable entire IDE block. |
| Spare | 7:1 | 00h | Scratch bits |
| AltDebugBusCntrl register | | | |

| C2Count - R – 8 bits - [PM_Reg: B3h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| C2Count | 7:0 | 00h | The value shows the amount of time the CPU spends in C2. Each increment represents approximately 0.39% (1/256). This register is updated by HW automatically every second |
| C2Count register | | | |

| C3Count - R – 8 bits - [PM_Reg: B4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| C3Count | 7:0 | 00h | The value shows the amount of time the CPU spends in C3. Each increment represents approximately 0.39% (1/256). This register is updated by HW automatically every second |
| C3Count register | | | |

| SmiIndicator0 - R – 8 bits - [PM_Reg: C0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiIndicator0 | 6:0 | - | This register is a read only register and is meant to help BIOS to search the SMI source quicker<br>Bit [0] = 1 means the SMI source is coming from PM_Reg0F<br>Bit [1] = 1 means the SMI source is coming from CMS (IO C50/C51, Index02)<br>Bit [2] = 1 means the SMI source is coming from PM_RegA9<br>Bit [3] = 1 means the SMI source is coming from PM_Reg1D<br>Bit [4] = 1 means the SMI source is coming from PM_Reg07<br>Bit [5] = 1 means the SMI source is coming from PM_Reg06<br>Bit [6] = 1 means the SMI source is coming from PM_Reg05<br>Bit [7] = 1 means the SMI source is coming from PM_Reg01 |
| Reserved | 7 | 0b | |
| SmiIndicator0 register | | | |

| SmiIndicator1 - R – 8 bits - [PM_Reg: C1h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiIndicator1 | 4:0 | - | This register is a read only register and is meant to help BIOS to search the SMI source quicker<br>Bit [0] = 1 means the SMI source is coming from PM_Reg5F<br>Bit [1] = 1 means the SMI source is coming from PM_Reg5D<br>Bit [2] = 1 means the SMI source is coming from PM_Reg5C<br>Bit [3] = 1 means the SMI source is coming from PM_Reg5B<br>Bit [4] = 1 means the SMI source is coming from PM_Reg5A |
| Reserved | 7:5 | 000b | |
| SmiIndicator1 register | | | |

| SmiIndicator2 - R – 8 bits - [PM_Reg: C2h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiIndicator2 | 5:0 | - | This register is a read only register and is meant to help BIOS to search the SMI source quicker<br>Bit [0] = 1 means the SMI source is coming from HWM (PMIO2)<br>Bit [1] = 1 means the SMI source is coming from AMDSI (PMIO2)<br>Bit [2] = 1 means the SMI source is coming from the fan monitor (PMIO2)<br>Bit [3] = 0 means the SMI source is coming from RTC VRT (IO 70/71, index 0D, bit 7). This bit is inverted.<br>Bit [4] = 1 means the SMI source is coming from SCI<br>Bit [5] = 1 means the SMI source is coming from ACPI_GEVENT_STATUS |
| Reserved | 7:6 | 00b | |
| SmiIndicator2 register | | | |

#### *2.3.3.3 ACPI Registers*

| Register Name | Offset Address* |
|---|---|
| Pm1Status | 00h |
| Pm1Enable | 02h |
| PmControl | 00h |
| PmaControl | 00h |
| TmrValue/ETmrValue | 00h |
| CLKVALUE | 00h |
| PLvl2 | 04h |
| PLvl3 | 05h |
| PLvl4 | 06h |
| AcpiSsCnt | 00h |
| EVENT_STATUS | 00h |
| EVENT_ENABLE | 04h |
| *** Note**: The offset addresses listed here for the ACPI registers belong to different apertures/decodes. Check the register descriptions for details. | |

| Pm1Status - RW - 16 bits - [AcpiPmEvtBlk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TmrStatus | 0 | 0b | Timer carry status bit. This bit gets set anytime the 23rd/31st bit of 24/32 bit counter changes (whenever the MSB changes from low to high or high to low. While TmrEn and TmrStatus are set, an interrupt event is raised). [Read-only] |
| Reserved | 3:1 | 000b | |
| BmStatus | 4 | 0b | Bus master status bit. This bit is set any time a system bus master requests the system bus, and can only be cleared by writing an one to this bit position. |
| GblStatus | 5 | 0b | This bit is set when an SCI is generated due to the BIOS wanting the attention of the SCI handler. This is set by writing 1 to PM_Reg: 0Eh bit [1]. |
| Reserved | 7:6 | 00b | |
| PwrBtnStatus | 8 | 0b | Power button status bit |
| Reserved | 9 | 0b | |
| RtcStatus | 10 | 0b | This bit is set when RTC generates an alarm. |
| Reserved | 13:11 | 000b | |
| PciExpWakeStatus | 14 | 0b | This bit is set by hardware to indicate that the system woke due to a PCI Express wakeup event. |
| WakeStatus | 15 | 0b | This bit is set when the system is in the sleep state and a wake-up event occurs. |
| This register is located at the base address defined by AcpiPmEvtBlk. | | | |

| Pm1Enable - RW - 16 bits - [AcpiPmEvtBlk:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TmrEn | 0 | 0b | This is the timer carry interrupt enable bit. When this bit is set then an SCI event is generated anytime the TmrStatus is set. When this bit is reset then no interrupt is generated when the TmrStatus bit is set. |
| Reserved | 4:1 | 0h | |
| GblEn | 5 | 0b | If this bit is set, SCI is raised whenever both GblEn and GblStatus are true. |
| Reserved | 7:6 | 00b | |
| PwrBtnEn | 8 | 0b | If this bit is set, SCI is generated whenever PwrBtnStatus is true. |
| Reserved | 9 | 0b | |
| RtcEn | 10 | 0b | RTC enable. If this bit is set, SCI is generated whenever RtcStatus is true. |

| Pm1Enable - RW - 16 bits - [AcpiPmEvtBlk:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 13:11 | 000b | |
| PciExpWakeDis | 14 | 1b | This bit disables the inputs to the PciExpWakeStatus from waking the system. |
| Reserved | 15 | 0b | |
| This register is located at the base address defined by AcpiPmEvtBlk. | | | |

| PmControl - RW - 16 bits - [AcpiPm1CntBlk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SCI_EN | 0 | 0b | Selects the power management event to be either an SCI or SMI# interrupt for the following events. When this bit is set, then PM events will generate an SCI interrupt; otherwise, it will be SMI#. |
| BmRld | 1 | 0b | If this bit is set, SCI is raised whenever there is a bus master active |
| GBL_RLS | 2 | 0b | If PM IO x0E bit[0] is set, writing 1 to this bit will generate SMI# and set PM IO x0F bit[0]. This bit will always return 0. |
| Reserved | 9:3 | 00h | |
| SlpType | 12:10 | 000b | Defines the sleep state the system enters when the SlpEn is set to one. This design currently implements 3 states: S1, S3, and S5 |
| SlpEn | 13 | 0b | This is a write-only bit and reads from it always return zero. If PM_Reg 04h bit7 SLP_SMI_EN is 0, setting this bit causes the system to sequence into the sleeping state associated with the SlpType fields programmed. If SLP_SMI_EN is 1, setting this bit causes SMI#. Writing 0 to this bit has no effect. This applies to both P4 and K8 systems. |
| Reserved | 15:14 | 00b | |
| This register is located at the base address defined by AcpiPm1CntBlk. | | | |

| PmaControl - RW - 8 bits - [AcpiPmaCntBlk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ARB_DIS | 0 | 0b | System arbiter is disabled when this bit is set. |
| Reserved | 7:1 | 00h | |
| This register is located at the base address defined by AcpiPmaCntBlk. | | | |

| TmrValue/ETmrValue – R - 32 bits - [AcpiPmTmrBlk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TmrValue | 31:0 | - | This read-only field returns the running count of the power management timer. |
| This register is located at the base address defined by AcpiPmTmrBlk. | | | |

| CLKVALUE - RW - 32 bits - [CpuControl:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| ClkValue | 3:1 | 000b | These bits define throttle interval for STPCLK# de-assertion<br>000b: 50%<br>001b: 12.5%<br>010b: 25%<br>011b: 37.5%<br>100b: 50%<br>101b: 62.5%<br>110b: 75%<br>111b: 87.5% |
| ThtEn | 4 | 0b | This bit enables clock throttling as set in the ClkValue. |
| Reserved | 31:5 | 0000000h | |

| CLKVALUE - RW - 32 bits - [CpuControl:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| This register is located at the base address defined by CpuControl | | | |

| PLvl2 - R - 8 bits - [CpuControl:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PLvl2 | 7:0 | 00h | Reads to this register return all zeros; writes to this register have no effect. Reads to this register generates a “enter C2 power” to the clock control logic (STPCLK logic). |
| This register is located at the base address defined by CpuControl | | | |

| PLvl3 – R – 8 bits - [CpuControl:05h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PLvl3 | 7:0 | 00h | Reads to this register return all zeros; writes to this register have no effect. Reads to this register generates a “enter C3 power” to the clock control logic (STPCLK logic). |
| This register is located at the base address defined by CpuControl | | | |

| PLvl4 – R - 8 bits - [CpuControl:06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PLvl4 | 7:0 | 00h | Reads to this register return all zeros; writes to this register have no effect. Reads to this register generates a “enter C4 power” to the clock control logic (STPCLK logic). |
| This register is located at the base address defined by CpuControl | | | |

| AcpiSsCnt - RW - 8 bits - [AcpiSsCntBlk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AcpiSsCnt | 0 | 0b | Reserved |
| Reserved | 7:1 | 00h | |
| This register is located at the base address defined by AcpiSsCntBlk | | | |

| EVENT_STATUS - RW - 32 bits - [AcpiGpe0Blk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GeventStatus | 7:0 | 00h | These bits indicate the status of the eight general purpose event signals events to the SB |
| LEventStatus | 8 | 0b | This bit indicates the status of the legacy power management logic implemented inside the SB. |
| TwarnStatus | 9 | 0b | This bit indicates the Temperature Caution input. |
| Reserved | 10 | | |
| USBStatus | 11 | 0b | This bit indicates the PME# from the internal USB controller |
| AC97Status | 12 | 0b | This bit indicates the PME# from the internal ac97 controller |
| OtherThermStatus | 13 | 0b | This bit indicates the status of OtherTherm from NB, fan, etc. |
| GPM9Status | 14 | 0b | This bit indicates the status of GPM[9] to SCI/Wakeup |
| PCIeHotPlugStatus | 15 | 0b | This bit indicates the status of PCIeHotPlug |
| ExtEvent0Status | 16 | 0b | This bit indicates the status of ExtEvent0 to SCI/Wakeup |
| ExtEvent1Status | 17 | 0b | This bit indicates the status of ExtEvent1 to SCI/Wakeup |
| PCIePmeStatus | 18 | 0b | This bit indicates the PME# from PCIExpress |
| GPM0Status | 19 | 0b | This bit indicates the status of GPM[0] to SCI/Wakeup |
| GPM1Status | 20 | 0b | This bit indicates the status of GPM[1] to SCI/Wakeup |
| GPM2Status | 21 | 0b | This bit indicates the status of GPM[2] to SCI/Wakeup |
| GPM3Status | 22 | 0b | This bit indicates the status of GPM[3] to SCI/Wakeup |
| GPM8Status | 23 | 0b | This bit indicates the status of GPM[8] to SCI/Wakeup |
| Gpio0Status | 24 | 0b | This bit indicates the status of GPIO0 (or WAKE#/GEVENT8 pin if PM IO Reg 84h bit1 =1) to SCI/wakeup |
| GPM4Status | 25 | 0b | This bit indicates the status of GPM[4] to SCI/Wakeup |
| GPM5Status | 26 | 0b | This bit indicates the status of GPM[5] to SCI/Wakeup |

| EVENT_STATUS - RW - 32 bits - [AcpiGpe0Blk:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AzaliaStatus | 27 | 0b | This bit indicates the status from the internal HD Audio controller |
| GPM6Status | 28 | 0b | This bit indicates the status of GPM[6] to SCI/Wakeup |
| GPM7Status | 29 | 0b | This bit indicates the status of GPM[7] to SCI/Wakeup |
| A11: Gpio2Status<br>A12:<br>Gpio2Status/Gpio66Status | 30 | 0b | A11: This bit indicates the status of GPIO2 to SCI/wakeup<br>A12: This bit indicates the status of GPIO2 or GPIO66; depending on PMIO_10h[7]. If PMIO_10h[7]=0, then it is routed to GPIO2. If PMIO_10h[7]=1, then it is GPIO66 |
| SataSciStatus | 31 | 0b | This bit indicates the status of SataSci to SCI/wakeup |
| This register is located at the base address defined by AcpiGpe0Blk. | | | |

| EVENT_ENABLE - RW - 32 bits - [AcpiGpe0Blk:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GeventEnable | 7:0 | 00h | These bits enable the GeventStatus to SCI output. |
| LeventEnable | 8 | 0b | This bit enables LEventStatus to SCI generation. |
| TwarnEnable | 9 | 0b | This bit enables TwarnStatus to SCI generation. |
| Reserved | 10 | 0b | |
| USBEnable | 11 | 0b | This bit enables PME# from the internal USB controllers |
| AC97Enable | 12 | 0b | This bit enables PME# from the internal ac97 controllers |
| OtherThermEnable | 13 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 13 OtherThermStatus |
| GPM9Enable | 14 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 14 GPM9Status |
| PCIeHotPlugEnable | 15 | 0b | Enable bit for AcpiGpe0Blk, offset 00, bit 15 PCIeHotPlugStatus |
| ExtEvent0Enable | 16 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 16 ExtEvent0Status |
| ExtEvent1Enable | 17 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 17 ExtEvent1Status |
| PCIePmeEnable | 18 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 18 PCIePmeStatus |
| GPM0Enable | 19 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 19 GPM0Status |
| GPM1Enable | 20 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 20 GPM1Status |
| GPM2Enable | 21 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 21 GPM2Status |
| GPM3Enable | 22 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 22 GPM3Status |
| GPM8Enable | 23 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 23 GPM8Status |
| Gpio0Enable | 24 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 24 Gpio0Status |
| GPM4Enable | 25 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 25 GPM4Status |
| GPM5Enable | 26 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 26 GPM5Status |
| AzaliaEnable | 27 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 27; for the internal HD Audio PME |
| GPM6Enable | 28 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 28 GPM6Status |
| GPM7Enable | 29 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 29 GPM7Status |
| Gpio2Enable | 30 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 30 Gpio2Status |
| SataSciEnable | 31 | 0b | Enables bit for AcpiGpe0Blk, offset 00, bit 31 SataSciStatus |
| This register is located at the base address defined by AcpiGpe0Blk. | | | |

| SmiCmdPort - RW - 8 bits – [SmiCmdBlk: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiCmdPort | 7:0 | 00h | Used by BIOS and OS |
| This register is located at the base address defined by AcpiSmiCmd + offset 0. | | | |

| SmiCmdStatus - RW - 8 bits – [SmiCmdBlk: 01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SmiCmdStatus | 7:0 | 00h | Used by BIOS and OS |

| SmiCmdStatus - RW - 8 bits – [SmiCmdBlk: 01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| This register is located at the base address defined by AcpiSmiCmd + offset 1. | | | |

### *2.3.4 WatchDogTimer Registers*

WatchDogTimer base address is defined in PM_Reg 6F:6Ch.

| Register Name | Offset Address |
|---|---|
| WatchDogControl | 00h |
| WatchDogCount | 04h |

| WatchDogControl - RW - 32 bits - [WD_Mem_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogRunStopB | 0 | 0b | This bit is used to control or indicate whether the watchdog is in the Running and Stopped states.<br>1 = Watchdog is in the Running state<br>0 = Watchdog is in the Stopped state<br>If the watchdog is in the Stopped state and a 1 is written to bit 0, the watchdog moves to the Running state, but a count interval is not started until a 1 is written to bit 7.<br>If the watchdog is in the Running state, writing a 1 to bit 0 has no effect.<br>The bit is only valid when the watchdog is enabled. |
| WatchDogFired | 1 | 0b | A value of "1" indicates that the watchdog timer has expired and caused the current restart. The bit is cleared by writing a "1" to bit 1 in the Watchdog Control register. Writing a "0" has no effect.<br>The bit is cleared by a power cycle or by the operating system and it must remain cleared for any restart that is not caused by the watchdog timer firing.<br>The bit is only valid when the watchdog is enabled. |
| WatchDogAction | 2 | 0b | This bit determines the action to be taken when the watchdog timer expires.<br>0=system reset<br>1=system power off<br>The bit is only valid when the watchdog is enabled. |
| WatchDogDisable | 3 | Same as PM IO x69, bit 0 | This bit reflects the state of the watchdog timer hardware.<br>0=Enable<br>1=Disable |
| Reserved | 6:4 | 000b | |
| WatchDogTrigger (WO) | 7 | 0b | Write only. Setting this bit triggers the watchdog to start a new count interval, counting down from the value that was last written to the Watchdog Count Register.<br>This bit is always read as zero. Setting this bit has no effect if the watchdog is disabled or stopped. |
| Reserved | 31:8 | 000000h | |

| WatchDogCount - RW - 32 bits - [WD_Mem_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WatchDogCount | 15:0 | ---- | This defines the countdown time for the counter. The units are defined in the Units field in the Watchdog Resource Table (WDRT). The maximum value is defined in the Max Count field in the WDRT.<br>Reading this register returns in the current counter value. |
| Reserved | 31:16 | 0000h | |

### *2.3.5 ASF SM bus Host Interface Registers*

The ASF SM bus host register block is resident in the Io space whose base defined at offset 58h/59h of config space.

| Register Name | Offset Address |
|---|---|
| HostStatus | 00h |
| HostControl | 02h |
| HostCommand | 03h |
| SlaveAddress | 04h |
| Data0 | 05h |
| Data1 | 06h |
| Data | 07h |
| PEC | 08h |
| ASFStatus | 0Ah |
| StatusMask0 | 0Bh |
| StatusMask1 | 0Ch |
| SlaveControl | 0Dh |
| RemoteCtrlAdr | 0Eh |
| SensorAdr | 0Fh |

| HostStatus – R - 8 bits - [ASF_IO: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| HostBusy | 0 | 0b | 0 – SM bus Host is idle<br>1 – SM bus Host is busy |
| Reserve | 1 | 0b | |
| DevError | 2 | 0b | 0: Slave device behave correctly<br>1: No ACK or Slave device responses incorrectly |
| BusCollision | 3 | 0b | 0: No bus collision<br>1: Bus collision |
| PECError | 4 | 0b | 0: No CRC error<br>1: CRC error happens |
| Reserve | 6:5 | 00b | |
| LastByte | 7 | 0b | 0: Last byte has not received<br>1: Last byte has received |

| HostControl – RW - 8 bits - [ASF_IO: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| KillHost | 1 | 0b | 0 – Enable SM master<br>1 – Reset SM master |
| Protocol | 4:2 | 000b | 000: Quick<br>001: Byte<br>010: Byte Data<br>011: Word Data<br>100: Process call<br>101: Block |
| PECAppend | 5 | 0b | 0: No PEC append<br>1: Automatic PEC append. ASF HC calculates CRC code and append to the tail of the data packets. |
| Start | 6 | 0b | WO:<br>0: Always read 0 on reads<br>1: Writing 1 to initiate the command |

| HostControl – RW - 8 bits - [ASF_IO: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PECEnable | 7 | 0b | 0: PEC disable<br>1: PEC enable, enable CRC checking when ASF HC presents as SM master and SM slave. |

| HostCommand – RW - 8 bits - [ASF_IO: 03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| HostCommand | 7:0 | 00h | Command to be transmitted by master |

| SlaveAddress– RW - 8 bits - [ASF_IO: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RW | 0 | 0b | 0: Write<br>1: Read |
| Address | 7:1 | 00h | Provide the SM address of Slave |

| Data0– RW - 8 bits - [ASF_IO: 05h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Data0 | 7:0 | 00h | Contains count or DATA0 field of transaction |

| Data1– RW - 8 bits - [ASF_IO: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Data1 | 7:0 | 00h | Contains DATA1 field of transaction |

| DataIndex– RW - 8 bits - [ASF_IO: 07h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DataIndex | 7:0 | 00h | Index to 32 Data registers. |

| PEC– RW - 8 bits - [ASF_IO: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PEC | 7:0 | 00h | PEC byte to be sent to slave. |

| ASFStatus– RW - 8 bits - [ASF_IO: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| EnableStatus | 7 | 0b | 1: Reset all the status bit in this register.<br>0: Enable status bit in this register |
| Reserved | 6:4 | 000b | |
| RemotePowerCycle | 3 | 0b | Power cycle has happened from ASF |
| RemotePowerUp | 2 | 0b | Power up has happened. from ASF |
| RemotePowerDown | 1 | 0b | Power down has happened from ASF |
| RemoteReset | 0 | 0b | Reset has happened from ASF |

| StatusMask0– RW - 8 bits - [ASF_IO: 0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Temp0StatusEnable | 0 | 0b | 1: Report Temp0 status to ASF<br>0: No report |
| Temp1StatusEnable | 1 | 0b | 1: Report Temp1 status to ASF<br>0: No report |
| Temp2StatusEnable | 2 | 0b | 1: Report Temp2 status to ASF<br>0: No report |
| Temp3StatusEnable | 3 | 0b | 1: Report Temp3 status to ASF<br>0: No report |
| AMDSIStatusEnable | 4 | 0b | 1: Report AMDSI status to ASF<br>0: No report |
| FanSpeed0StatusEnable | 5 | 0b | 1: Report Fan0 Speed Status to ASF<br>0: No report |
| FanSpeed1StatusEnable | 6 | 0b | 1: Report Fan1 Speed Status to ASF<br>0: No report |
| FanSpeed2StatusEnable | 7 | 0b | 1: Report Fan2 Speed Status to ASF<br>0: No report |

| StatusMask1– RW - 8 bits - [ASF_IO: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AnalogIo0StatusEnable | 0 | 0b | 1: Report AnalogIo0 status to ASF<br>0: No report |
| AnalogIo1StatusEnable | 1 | 0b | 1: Report AnalogIo1 status to ASF<br>0: No report |
| AnalogIo2StatusEnable | 2 | 0b | 1: Report AnalogIo2 status to ASF<br>0: No report |
| AnalogIo3StatusEnable | 3 | 0b | 1: Report AnalogIo3 status to ASF<br>0: No report |
| AnalogIo4StatusEnable | 4 | 0b | 1: Report AnalogIo4 status to ASF<br>0: No report |
| AnalogIo5StatusEnable | 5 | 0b | 1: Report AnalogIo5 status to ASF<br>0: No report |
| AnalogIo6StatusEnable | 6 | 0b | 1: Report AnalogIo6 status to ASF<br>0: No report |
| AnalogIo7StatusEnable | 7 | 0b | 1: Report AnalogIo7 status to ASF<br>0: No report |

| SlaveMisc- RW - 8 bits - [ASF_IO: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SlavePECError | 0 | 0b | RO<br>0: No PEC error<br>1: PEC error |
| SlaveBusCollision | 1 | 0b | RO<br>0: No BusCollision<br>1: BusCollision happens |
| SlaveDevError | 2 | 0b | RO<br>0: Expected response<br>1: Unexpected response |
| WrongSP | 3 | 0b | RO<br>0: No SP error<br>1: No SP when turn to read |
| Reserved | 4 | 0b | |
| SuspendSlave | 5 | 0b | RW<br>Write 1 to Suspend (stop) ASF Slave state machine |

| SlaveMisc- RW - 8 bits - [ASF_IO: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| KillSlave | 6 | 0b | RW<br>Write 1 to reset Slave ASF Slave state machine |
| LegacySensorEn | 7 | 0b | RW<br>0: Disable Legacy Sensor<br>1: Enable Legacy Sensor |

| RemoteCtrlAdr– RW - 8 bits - [ASF_IO: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| RemoteCtrlAdr | 7:1 | 00h | SM address of Remote Control device. |

| SensorAdr– RW - 8 bits - [ASF_IO: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| SensorAdr | 7:1 | 00h | SM address of Sensor. |

The SB600 is hardcoded to recognize the following ASF commands. The BIOS should report these values accordingly in the ASF table if the system supports ASF function.

| | Control command | Control data value |
|---|---|---|
| Reset | 00h | 00h |
| PowerUp | 01h | 00h |
| PowerDown | 02h | 00h |
| PowerCycle | 03h | 00h |
| SensorPolling | 23h | 00h |

## 2.4 IDE Controller (Device 20, Function 1)

### *2.4.1 PCI Configuration Registers*

| Register Name | Offset Address |
|---|---|
| Vendor ID | 00h |
| Device ID | 02h |
| Command | 04h |
| Status | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Master Latency Timer | 0Dh |
| Header Type | 0Eh |
| BIST Mode Type | 0Fh |
| Base Address 0 | 10h |
| Base Address 1 | 14h |
| Base Address 2 | 18h |
| Base Address 3 | 1Ch |
| Bus Master Interface Base Address | 20h |
| Subsystem ID and Subsystem Vendor ID | 2Ch |
| Capabilities Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Min_gnt | 3Eh |
| Max_latency | 3Fh |
| IDE PIO Timing | 40h |
| IDE Legacy DMA (Multi-words DMA) Timing Modes | 44h |
| IDE PIO Control | 48h |
| IDE PIO Mode | 4Ah |
| IDE Status | 4Ch |
| IDE Ultra DMA Control | 54h |
| IDE Ultra DMA Status | 55h |
| IDE Ultra DMA Mode | 56h |
| IDE PCI Retry Timing Counter | 60h |
| PCI Error Control | 61h |
| IDE Internal Control | 62h |
| IDE Misc. Control | 64h |
| IDE MSI Programmable Weight | 68h |
| IDE Dynamic Clocking | 6Ch |
| IDE MSI Control | 70h |
| IDE MSI Address Register | 74h |
| IDE MSI Data Register | 78h |

| Vendor ID - R - 16 bits - [PCI_Reg:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | This register holds a unique 16-bit value assigned to a vendor, and combined with the device ID, it identifies any PCI device. |
| Vendor ID Register: This register holds a unique 16-bit value assigned to a vendor, and combined with the device ID it identifies any PCI device. | | | |

| Device ID - R - 16 bits - [PCI_Reg:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID | 15:0 | 438Ch | This register holds a unique 16-bit value assigned to a device, and combined with the vendor ID it, identifies any PCI device. |

| Device ID - R - 16 bits - [PCI_Reg:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID Register: This register holds a unique 16-bit value assigned to a device, and combined with the vendor ID it identifies any PCI device. | | | |

| Command - RW - 16 bits - [PCI_Reg:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| I/O Access Enable | 0 | 0b | I/O Access Enable. This bit controls access to the I/O space registers. When this bit is 1, it enables access to Legacy IDE ports, and PCI bus master IDE I/O registers are enabled. |
| Memory Access Enable | 1 | 0b | Memory Access Enable. This function is not implemented. This bit is always 0. |
| Bus Master Enable | 2 | 0b | Master Enable. Bus master function enable.<br>1 = Enable<br>0 = Disable. |
| Special Cycle Recognition Enable | 3 | 0b | Special Cycle recognition enable. This feature is not implemented and this bit is always 0. |
| Memory Write and Invalidate Enable | 4 | 0b | Memory Write and Invalidate Enable. |
| VGA Palette Snoop Enable | 5 | 0b | VGA Palette Snoop Enable- The IDE host controller does not need to snoop VGA palette cycles. This bit is always 0. |
| PERR- Detection Enable | 6 | 0b | PERR- (Response) Detection Enable bit - If set to 1, the IDE host controller asserts PERR- when it is the agent receiving data AND it detects a parity error. PERR- is not asserted if this bit is 0.<br>Default - 0. |
| Wait Cycle Enable | 7 | 0b | Wait Cycle enable - The IDE host controller does not need to insert a wait state between the address and data on the AD lines. This bit is always 0. |
| SERR- Enable | 8 | 0b | SERR- enable - If set to 1, the IDE host controller asserts SERR- when it detects an address parity error. SERR- is not asserted if this bit is 0.<br>Default - 0. |
| Fast Back-to-Back Enable | 9 | 0b | Fast Back-to-back enable. The IDE host controller only acts as a master to a single device, so this functionality is not needed. This bit is always 0. |
| Interrupt Disable | 10 | 0b | Interrupt disable bit (comply to PCI 2.3 spec.) |
| Reserved | 15:11 | 00h | Reserved. Always wired as 0's. |
| Command Register: The PCI specification defines this register to control a PCI device's ability to generate and respond to PCI cycles. | | | |

| Status - RW - 16 bits - [PCI_Reg:06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 0b | Reserved. These bits are always read as 0. |
| Interrupt Status | 3 | 0b | Interrupt status bit. It complies with the PCI 2.3 specification. |
| Capabilities List | 4 | 0b | This bit is enabled by PCI Config offset 0x62 bit 13 to indicate that the Capabilities Pointer is located at 34h. |
| 66MHz Support | 5 | 1b | 66MHz capable. This feature is supported in the IDE host controller. |
| UDF Supported | 6 | 0b | UDF Supported. This feature is not implemented and this bit is always 0. |
| Fast Back-to-Back Capable | 7 | 0b | Fast Back-to-Back Capable. This feature is not implemented and this bit is always 0. |
| Data Parity Error | 8 | 0b | Data Parity reported – Set to 1 if the IDE host controller detects PERR- asserted while acting as PCI master (whether PERR- was driven by IDE host controller or not.) |

| Status - RW - 16 bits - [PCI_Reg:06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DEVSEL- Timing | 10:9 | 01b | DEVSEL- timing – Read only bits indicating DEVSEL- timing when performing a positive decode.<br>Since DEVSEL- is asserted to meet the medium timing, these bits are encoded as 01b. |
| Signaled Target Abort | 11 | 0b | Signaled Target Abort – This bit is set to 1, when the IDE host controller signals Target Abort. |
| Received Target Abort | 12 | 0b | Received Target Abort – This bit is set to 1 when the IDE host controller-generated PCI cycle (IDE host controller is the PCI master) is aborted by a PCI target. Cleared by writing a 1 to it. |
| Received Master Abort Status | 13 | 0b | Received Master Abort Status. Set to 1 when the IDE host controller acting as a PCI master, aborts a PCI bus memory cycle. Cleared by writing a 1 to this bit.<br>Default - 0. |
| SERR- Status | 14 | 0b | SERR- status. This bit is set to 1 when the IDE host controller detects a PCI address parity error. |
| Detected Parity Error | 15 | 0b | Detected Parity Error. This bit is set to 1 when the IDE host controller detects a parity error. |
| Status Register: The PCI specification defines this register to record status information for PCI related events. This is a read/write register. However, writes can only reset bits. A bit is reset when the register is written and the data in the corresponding bit location is a 1. | | | |

| Revision ID/Class Code- RW - 32 bits - [PCI_Reg:08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | These bits are hardwired to 00h to indicate the revision level of the chip design (for the SB600). |
| IDE Host Controller Operating Mode Selection | 15:8 | 8Ah | Programmable I/F. These 8 bits are read/write.<br>Bit 7 – Master IDE Device. Always 1.<br>Bit 6-4 – Reserved. Always read as 0's.<br>Bit 3 – Secondary IDE Enable bit. Setting this bit with PCI configuration offset 0x49 bit 8 back to back will disable secondary IDE host controller.<br>Bit 2 – Operating Mode for Secondary.<br>1 = Native PCI-mode.<br>0 = Compatibility Mode(Default).<br>Bit 1 – Primary IDE Enable bit. Setting this bit with PCI configuration offset 0x49 bit 0 back to back will disable primary IDE host controller.<br>Bit 0 – Operating Mode for Primary.<br>1 = Native PCI-mode.<br>0 = Compatibility mode (Default). |
| Sub-Class Code | 23:16 | 01h | Sub-Class Code. These 8 bits are read only and wired to 01h to indicate an IDE Controller. |
| Class Code | 31:24 | 01h | Class Code. These 8 bits are read only and wired to 01h to indicate a Mass-Storage Controller. |
| Revision ID/Class Code Register: This register contains the device's revision information, generic function of a device, and the specific register level programming interface. The Base class is 01h (Mass-Storage Controller), Sub-class is 01h (IDE Controller). | | | |

| Cache Line Size - RW - 8 bits - [PCI_Reg:0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size Register | 7:0 | 00h | If the value is 10 that means the cache line size is 16 DW (64 byte). |
| Cache Line Size Register: This register specifies cache line size and the default value is 00. | | | |

| Master Latency Timer - RW - 8 bits - [PCI_Reg:0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 0h | They are not used and wired to 0. |
| Master Latency Timer | 7:3 | 00h | Master Latency Timer. This number represents the guaranteed time slice allotted to IDE host controller for burst transactions. |
| Master Latency Timer: This register specifies the value of Latency Timer in units of PCICLKs. | | | |

| Header Type - R - 8 bits - [PCI_Reg:0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 00h | Header Type. Since the IDE host controller is a single-function device, this register contains a value of 00h. |
| Header Type Register: This register identifies the IDE controller module as a single function device. | | | |

| BIST Mode Type - R - 8 bits - [PCI_Reg:0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Built-in-Self Test Mode | 7:0 | 00h | Built-in-Self Test modes. Since the IDE host controller does not support BIST modes, this register is always read as 00. |
| BIST Mode Type Register: This register is used for control and status for Built-in-Self test. The IDE host controller has no BIST modes | | | |

| Base Address 0 - RW - 32 bits - [PCI_Reg:10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 2:1 | 00b | Reserved. Always read as 0's. |
| Primary IDE CS0 Base Address | 15:3 | 0000h | Base Address for Primary IDE Bus CS0. This register is used for native mode only. Base Address 0 is not used in compatibility mode. |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's. |
| Base Address 0 Register (Primary CS0): This register identifies the base address of a contiguous IO space of command register block for the primary channel. | | | |

| Base Address 1 - RW - 32 bits - [PCI_Reg:14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 1 | 0b | Reserved. Always read as 0's. |
| Primary IDE CS1 Base Address | 15:2 | 0000h | Base Address for Primary IDE Bus CS1. This register is used for native mode only. Base Address 1 is not used in compatibility mode. |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's. |
| Base Address 1 Register (Primary CS1): This register identifies the base address of a contiguous IO space of command register block for the primary channel. | | | |

| Base Address 2 - RW - 32 bits - [PCI_Reg:18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 2:1 | 00b | Reserved. Always read as 0's. |
| Secondary IDE CS0 Base Address | 15:3 | 0000h | Base Address for Secondary IDE Bus CS0. This register is used for native mode only. Base Address 2 is not used in compatibility mode. |

| Base Address 2 - RW - 32 bits - [PCI_Reg:18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's. |
| Base Address 2 Register (Secondary CS0): This register identifies the base address of a contiguous IO space of command register block for the primary channel. | | | |

| Base Address 3 - RW - 32 bits - [PCI_Reg:1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 1 | 0b | Reserved. Always read as 0's. |
| Secondary IDE CS1 Base Address | 15:2 | 0000h | Base Address for Secondary IDE Bus CS1. This register is used for native mode only. Base Address 3 is not used in compatibility mode. |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's. |
| Base Address 3 Register (Secondary CS1): This register identifies the base address of a contiguous IO space of command register block for the primary channel. | | | |

| Bus Master Interface Base Address - RW - 32 bits - [PCI_Reg:20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Resource Type Indicator | 0 | 1b | RTE (Resource Type Indicator). This bit is wired to 1 to indicate that the base address field in this register maps to I/O space. |
| Reserved | 3:1 | 0h | Reserved. Always read as 0's. |
| Bus Master Interface Register Base Address | 15:4 | 000h | Base Address for Bus Master interface registers and correspond to AD[15:4]. |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's. |
| Bus Master Interface Base Address Register: This register selects the base address of a 16-byte I/O space interface for bus-master functions. | | | |

| Subsystem ID and Subsystem Vendor ID – RW - 32 bits - [PCI_Reg:2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Subsystem Vendor ID |
| Subsystem ID | 31:16 | 0000h | Subsystem ID |
| Subsystem ID and Subsystem Vendor ID: This subsystem ID and subsystem Vendor ID register is write once and read only. | | | |

| MSI Capabilities Pointer - R - 8 bits - [PCI_Reg:34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 00h | The first pointer of Capability block<br>Note: the capabilities pointer (0x70) can be enabled by setting PCI config offset 0x62 bit 13 to 1 |
| MSI Capabilities Pointer Register: This register will show the PCI configuration register starting address and it is read only. | | | |

| Interrupt Line - RW - 8 bits - [PCI_Reg:3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | Identifies which input on the interrupt controller the function's PCI interrupt request pin (as specified in its Interrupt Pin register) is routed to |
| Interrupt Line Register: This register identifies which of the system interrupt controllers the device interrupt pin is connected to. The value of this register is used by device drivers. | | | |

| Interrupt Pin - R - 8 bits - [PCI_Reg:3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 01h | Hard-wired to 01h. |
| Interrupt Pin Register: This register identifies the interrupt pin a device uses. Since the IDE host controller uses IRQ14, this value is supposed to be 00. However, the IDE controller will generate the PCI interrupt INTA# signal on the PCI bus. Therefore, this pin register is set to 01h. | | | |

| Min_gnt - R - 8 bits - [PCI_Reg:3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minimum Grant | 7:0 | 00h | Hard-wired to 0's and always read as 0's. |
| Min_gnt Register: This register specifies the desired settings for how long of a burst the IDE host controller needs assuming a clock rate of 33MHz. The value specifies a period of time in units of ¼ microseconds. | | | |

| Max_latency - R - 8 bits - [PCI_Reg:3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Maximum Latency | 7:0 | 00h | Hard-wired to 0's and always read as 0's. |
| Max_latency Register: This register specifies the Maximum Latency time required before the IDE host controller as a bus-master can start an accesses. | | | |

| IDE PIO Timing - RW - 32 bits - [PCI_Reg:40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Slave Data Register Command Recovery Width | 3:0 | 9h | Slave Data register command recovery width for Primary IDE bus slave PIO device. |
| Primary Slave Data Register Command width | 7:4 | 9h | Slave Data register command width for Primary IDE bus slave PIO device. |
| Primary Master Data Register Command Recovery Width | 11:8 | 9h | Master Data register command recovery width for primary IDE bus Master PIO device. |
| Primary Master Data Register Command Width | 15:12 | 9h | Master Data register command width for Primary IDE bus Master PIO device. |
| Reserved | 31:16 | 0000h | Reserved. Always read as 0's |

IDE PIO Timing Register: This register controls the IDE interface and selects the timing of the IDE PIO bus-master cycles.
**Note**: Relation of setting value and actual timing of each mode are

| PIO Mode | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Command Width | 2(90ns) | 2(90ns) | 3(120ns) | 4(150ns) | 9(270ns) |
| Recover Width | 0(30ns) | 2(90ns) | 4(150ns) | 7(240ns) | 9(270ns) |

The above timings are valid and A-Link clock is always 66MHz.
Actual timing is setting value + 1 A-Link clock cycle.

| IDE Legacy DMA (Multi-words DMA) Timing Modes - RW - 32 bits - [PCI_Reg:44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Slave DMA Command Recovery Width | 3:0 | Fh | Slave DMA command recovery width for Primary IDE bus Slave DMA device. |
| Primary Slave DMA Command Width | 7:4 | Fh | Slave DMA command width for Primary IDE bus Slave DMA device. |
| Primary Master DMA Command Recovery Width | 11:8 | Fh | Master DMA Command recovery width for primary IDE bus Master DMA device. |
| Primary Master DMA Command Width | 15:12 | Fh | Master DMA Command width for Primary IDE bus Master DMA device. |
| Reserved | 31:16 | 0h | Reserved. Always read as 0's |

| IDE Legacy DMA (Multi-words DMA) Timing Modes - RW - 32 bits - [PCI_Reg:44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE Legacy DMA (Multi-words DMA) Timing Modes Register: This register controls the IDE interface and selects the timing of the IDE DMA bus-master cycles.<br>**Note**: Relation of setting value and actual timing of each mode are<br>DMA Mode 2 1 0<br>Command Width 2(90ns) 2(90ns) 7(240ns)<br>Recover Width 0(30ns) 1(60ns) 7(240ns)<br>The above timings are valid and A-Link clock is always 66MHz.<br>Actual timing is setting value + 1 A-Link clock cycle. | | | |

| IDE PIO Control - RW - 16 bits - [PCI_Reg:48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary IDE Disable | 0 | 0b | Disable Primary IDE controller. When set Primary IDE controller is disabled. |
| Reserved | 3:1 | 000b | Reserved. Always read as 0's |
| Reserved | 8:4 | 00h | Obsolete bits |
| Reserved | 11:9 | 000b | Reserved. Always read as 0's |
| Reserved | 15:12 | 0h | Obsolete bits |
| IDE PIO Control Register: This register controls the IDE interface and selects the control functions of the PCI bus IDE PIO bus-master cycles. | | | |

| IDE PIO Mode - RW- 16 bits - [PCI_Reg:4Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Master PIO Access Mode | 2:0 | 0h | PIO access mode for Primary IDE Master device. For instance, PIO 0 = 000, PIO 1 = 001, … etc.. |
| Reserved | 3 | 0b | Reserved. Always read as 0's. |
| Primary Slave PIO Access Mode | 6:4 | 0h | PIO access mode for Primary IDE Slave device. For instance, PIO 0 = 000, PIO 1 = 001, … etc.. |
| Reserved | 15:7 | 000h | Reserved. Always read as 0's. |
| IDE PIO Mode Register: This register specifies PIO modes primary channel. | | | |

| IDE Status - R- 8 bits - [PCI_Reg:4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | Reserved. These bits always read as0's. |
| IDE Status Register: This register specifies the IDE Status of primary channel. | | | |

| IDE Ultra DMAControl - RW- 8 bits - [PCI_Reg:54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Master Ultra DMA enable | 0 | 0b | Ultra DMA Enable Primary IDE Master device. |
| Primary Slave Ultra DMA enable | 1 | 0b | Ultra DMA Enable Primary IDE Slave device. |
| Reserved | 6:2 | 00000b | Reserved. Always read as 0's. |
| Ultra DMA Report Mode | 7 | 0b | Report Mode.<br>When host receives last data as extra word, reporting will be:<br>If this bit is 1, set IDE Ultra DMA Status register only.<br>If this bit is 0, clear the interrupt bit of BusMaster Status Register and also set IDE Ultra DMA Status Register. |
| IDE Ultra DMA Control Register: This register specifies the IDE Control of primary channel. | | | |

| IDE Ultra DMA Status - RW- 8 bits - [PCI_Reg:55h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Extra Data Status | 0 | 0b | Transaction is complete, but internal buffer has some data. This bit will be cleared by resetting the DMA start bit. |
| Reserved | 7:1 | 00h | Reserved. Always read as 0's. |

| IDE Ultra DMA Status - RW- 8 bits - [PCI_Reg:55h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE Ultra DMA Status Register: This register specifies the Ultra DMA status for primary channel. | | | |

| IDE Ultra DMA Mode - RW- 16 bits - [PCI_Reg:56h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Master Ultra DMA Access Mode | 2:0 | 0h | Ultra DMA access mode for Primary IDE master device. For instance, UDMA-5 = 101, UDMA-6 = 110. |
| Reserved | 3 | 0b | Reserved. Always read as 0's. |
| Primary Slave Ultra DMA Access Mode | 6:4 | 0h | Ultra DMA access mode for Primary IDE Slave device. For instance, UDMA-5 = 101, UDMA-6 = 110. |
| Reserved | 15:7 | 000h | Reserved. Always wired as 0's. |
| IDE Ultra DMA Mode Register: This register specifies the Ultra DMA timings for primary channel. | | | |

| IDE PCI Retry Timing Counter - RW- 8 bits - [PCI_Reg:60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCI Retry Timing Counter | 3:0 | 0h | Waiting number of PCI clocks. If a PCI Master transaction is retried, the IDE host controller will wait for 0-15 PCI clocks, depending upon the value programmed, and will re-issue the PCI master transaction. |
| Reserved | 7:4 | 0h | Reserved. Always wired as 0's. |
| IDE PCI Retry Timing Counter Register: This register specifies the Ultra DMA re-try timings for primary channel. | | | |

| PCI Error Control - RW- 8 bits - [PCI_Reg:61h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCI Data Parity Check Ignore | 0 | 0b | 0 - Check parity bit for DMA transaction.<br>1 - Ignore parity bit for DMA transaction.<br>If there is a parity error with a DMA transaction, the IDE host controller does not set the error bit (Bus Master IDE Reg.) If this bit is 1, bit 1 is "don't care." (same as bit 1=0.) |
| PCI Descriptor DMA Abort Enable | 1 | 0b | 0 - DMA will not be aborted by PCI parity error at descriptor table read.<br>1- DMA will be aborted by PCI parity error at descriptor table read. |
| PCI Master/Target Abort Ignore | 2 | 0b | 0 - DMA will be aborted by parity error, master abort, and target abort at data transaction (if bit0=1 or bit1=0, DMA does not abort by parity error.)<br>1 - DMA does not abort by parity error, master abort, and target abort at data transaction. |
| Reserved | 7:3 | 00h | Reserved. Always wired as 0's. |
| PCI Error Control Register: This register specifies DMA parity bit errors. | | | |

| IDE Internal Control - RW- 16 bits - [PCI_Reg:62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE Internal PCI master request selection | 0 | 0b | Select number of delay cycles on internal PCI master request<br>0: delay two PCI cycles<br>1: delay three PCI cycles |
| IDE PCI request control | 1 | 0b | Control PCI request deassertion timing at PCI retry cycle.<br>0: allows IDE's PCI request to be deasserted one cycle early.<br>1: PCI request will be deasserted one cycle late. |
| Reserved | 5:2 | 0h | Reserved. Always read as 0's |
| IDE Fast sampling Enable | 6 | 1b | Enable fast sampling on IORDY (DDMARDY#) port<br>0: Use regular IDE clock to sample<br>1: Use either (50/66MHz) sampling IORDY (DDMARDY#) port at ultra DMA mode |
| Reserved | 7 | 0b | Reserved. |
| Explicit/Implicit prefetch Switch | 8 | 0b | Control implicit or explicit prefetch function<br>0: Explicit prefetch is enabled<br>1: Implicit prefetch is enabled<br>Recommendation is to have it reset to 0 for better bus efficiency |
| Reserved | 12:9 | 0h | Reserved |
| MSI capability visible control | 13 | 0b | When this bit is set MSI capability will be visible. |
| Reserved | 14 | 0b | Reserved |
| MSI_MME[0] hidden control bit | 15 | 0b | Control Multiple/Single Message Enable<br>0: Only 1 message allocated<br>1: 2 messages allocated |
| IDE Internal Control Register: This register can enable internal PCI request timing and different IDE arbitration schemes. | | | |

| IDE Misc. Control - RW- 32 bits - [PCI_Reg:64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | Reserved. |
| IDE Internal Logic Reset | 1 | 0b | IDE internal reset. This reset follows IDE PLL reset. Software control the duration between both resets. |
| Primary Channel Tri-state Mode | 2 | 0b | Primary channel tri-state mode if it is set to 1. |
| Reserved | 3 | 0b | Reserved. Always read as 0's. |
| Preliminary Buffer Threshold Control | 6:4 | 001b | Programmable threshold control for preliminary buffer. |
| Reserved | 31:7 | 00E0258h | Reserved. Do not use |
| IDE Misc. Control Register: This register specifies IDE tri-state channel enable, and preliminary buffer threshold control. | | | |

| IDE MSI Programmable Weight - RW- 8 bits - [PCI_Reg:68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Interrupt Weight | 5:0 | 01h | MSI programmable interrupt weight. |
| Reserved | 7:6 | 0h | Reserved. Always wired as 0's. |
| IDE MSI Programmable Weight Register: This register specifies MSI weight. | | | |

| IDE Dynamic Clocking - RW- 20 bits - [PCI_Reg:6Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE Power Down Counter | 19:0 | FFFFFh | The IDE power down counter can be programmed to shut down the IDE clock.<br>The counter is running at 66Mhz clock. If the value is set to 0x3FF, it means that the IDE controller detects that the bus is without activity for more than 15us, the internal clock is powered down for advance power saving. Any detected activity will turn the clock back on. The default is FFFFFh and dynamic clocking is disabled. |
| Reserved | 31:20 | 000h | Reserved. Always wired as 0's. |
| IDE Dynamic Clocking Register: This register specifies the time (in number of IDE clocks) to shut down the IDE clock after there are no more IDE transactions.<br>Note: A 32-bit double-word (DW) PCI configuration write is required to update this register. | | | |

| IDE MSI Control - RW- 32 bits - [PCI_Reg:70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 05h | Capability ID (hard_wired to 05h) |
| Capability Next Pointer | 15:8 | 00h | Next Pointer (hard_wired to 00h) |
| Message Signaled Interrupt Enable | 16 | 0b | MSI Enable (MSI_En) |
| Multiple Message Capable | 19:17 | 0h | Multiple Message Capable (MMC) |
| Multiple Message Enable | 22:20 | 0h | Multiple Message Enable (MME) (hard_wired to 0h) |
| MSI 64-bit Address | 23 | 0b | 64-bit address (hard_wired to 0b) |
| Reserved | 31:24 | 00h | Reserved. Always wired as 0's. |
| IDE MSI Control Register: This register specifies MSI Capability ID, next pointer, MSI enable, multiple message capable, multiple message enable bits. | | | |

| IDE MSI Address Register - RW- 32 bits - [PCI_Reg:74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE MSI Address | 31:0 | 0000_0000h | MSI Address |
| IDE MSI Address Register: This register specifies MSI address. | | | |

| IDE MSI Data Register - RW- 16 bits - [PCI_Reg:78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IDE MSI Data | 15:0 | 0000h | MSI Data |
| IDE MSI Data Register: This register specifies MSI data. | | | |

### *2.4.2 IDE I/O Registers*

The PCI IDE function uses 16 bytes of I/O space. These registers are accessed through the Bus-master interface base address register.

**Bus-master IDE Command Register**

Address Offset: Primary – Base + 00h

| Register Name | Offset Address [Primary] |
|---|---|
| Bus-master IDE Command | 00h/08h |
| Bus-master IDE Status | 02h/0Ah |
| Bus-master IDE Command | 04h/0Ch |

| Bus-master IDE Command - RW- 8 bits - [IDE:00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Bus Master IDE Start/Stop | 0 | 0b | Bus Master IDE Start (1)/Stop (0).<br>This bit will not be reset by interrupt from IDE device. This must be reset by soft ware (device driver). |
| Reserved | 2:1 | 0h | Reserved. Wired 0's. |
| Bus Master Read/Write | 3 | 0b | Bus Master IDE r/w (direction) control<br>0 = Memory -> IDE<br>1 = IDE -> Memory<br>This bit should not change during Bus Master transfer cycle, even if terminated by Bus Master IDE stop. |
| Reserved | 7:4 | 0h | Reserved. These bits are always read as 0's. |

**Bus-master IDE Status Register**

Address Offset: Primary – Base + 02h

| Bus-master IDE Status - RW- 8 bits - [IDE:02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Bus Master Active | 0 | 0b | Bus Master IDE active. This bit is set to 1 when bit 0 in the Bus Master IDE command address register is set to 1. The IDE host controller sets this bit to 0 when the last transfer for a region is performed. This bit is also set to 0 when bit 0 of the Bus Master IDE command register is set to 0. |
| Bus Master DMA Error | 1 | 0b | IDE DMA error. This bit is set when the IDE host controller encounters a target abort, master abort, or Parity error while transferring data on the PCI bus. Software sets this bit to a 0, by writing a 1 to it. |
| IDE Interrupt | 2 | 0b | IDE Interrupt. Indicates when an IDE device has asserted its interrupt line. IRQ14 is used for the primary channel. If the interrupt status bit is set to 0, by writing a 1 to this bit while the interrupt line is still at the active level, this bit remains 0 until another assertion edge is detected on the interrupt line. |
| Reserved | 4:3 | 0h | Reserved. Always read as 0's. |
| Master Device DMA Capable | 5 | 0b | Device 0 (Master) DMA capable. |
| Slave Device DMA Capable | 6 | 0b | Device 1 (Slave) DMA capable. |
| Simplex Only | 7 | 0b | Simplex only. This bit is hard-wired as 0. |

**Descriptor Table Pointer Register**

Address Offset: Primary – Base + 04h

| Bus-master IDE Command - RW- 32 bits - [IDE:04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 0h | Reserved. Always read as 0's. |
| Descriptor Table Base Address | 31:2 | 0000_0000h | Base Address of Descriptor Table. These bits correspond to Address [31-02]. |

**Table 2-8 IDE Device Registers Mapping**

| **Address (hex)** | | **Name and Function** | |
|---|---|---|---|
| **Compatibility Mode** | **Native Mode (Offset)** | **Read Function** | **Write Function** |
| **IDE Command Block Registers** | | | |
| **Primary** | | | |
| 1F0 | Base Address 0 + 0 | Data (16 bit) | Data (16 bit) |

| Address (hex) | | Name and Function | |
|---|---|---|---|
| **Compatibility Mode** | **Native Mode (Offset)** | **Read Function** | **Write Function** |
| 1F1 | Base Address 0 + 1 | Error register | Features register |
| 1F2 | Base Address 0 + 2 | Sector Count | Sector Count |
| 1F3 | Base Address 0 + 3 | Sector Number | Sector Number |
| 1F4 | Base Address 0 + 4 | Cylinder Low | Cylinder Low |
| 1F5 | Base Address 0 + 5 | Cylinder High | Cylinder High |
| 1F6 | Base Address 0 + 6 | Drive/Head | Drive/Head |
| 1F7 | Base Address 0 + 7 | Status | Command |
| **IDE Control Block Registers** | | | |
| **Primary** | | | |
| 3F6 | Base Address 1 + 2 | Alternate Status | Device Control |

## 2.5 AC '97 Controller Functional Descriptions

### *2.5.1 Audio Registers (Device 20, Function 5)*

The PCI based registers for Audio are defined according to the PCI 2.1 specification and Windows 2000 requirements.

#### *2.5.1.1 PCI Configuration Registers*

| Register Name | Offset Address |
|---|---|
| Vendor ID | 00h |
| Device ID | 02h |
| PCI Command | 04h |
| PCI Device Status | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| Built-in Self Test | 0Fh |
| Base Address Register 0 | 10h |
| Base Address Register 1 | 14h |
| Cardbus CIS pointer (Reserved) | 28h |
| Subsystem ID & Subsystem Vendor ID | 2Ch |
| Capabilities Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Min Grant | 3Eh |
| Max Latency | 3Fh |
| MSI Capability Register Set IDs | 40h |
| MSI Message Control | 42h |
| MSI Message Address | 44h |
| MSI Message Data | 48h |
| MSI Program Weight | 4Ch |
| UnMask Latency Timer Expiration | 50h |

**IMPORTANT: The driver is required to check revision ID to enable functions appropriately. AMD's AC'97 controller will be backward compatible to previous revisions. For example, revision 01 will have everything the same as revision 00, plus the enhancement**.

| Vendor ID- R - 16 bits - [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | Vendor ID |
| Vendor ID Register: This register holds a unique 16-bit value assigned to a vendor, and combined with the device ID it identifies any PCI device. | | | |

| Device ID- R - 16 bits - [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID | 15:0 | 4382h | Device ID |
| Device ID Register: This register holds a unique 16-bit value assigned to a device, and combined with the vendor ID it identifies any PCI device. | | | |

| CMD- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Space | 0 | 0b | I/O Access Enable. |
| Memory Space | 1 | 0b | Memory Access Enable. |
| Bus Master | 2 | 0b | Master Enable. |
| Special Cycles | 3 | 0b | Hardwired to 0 to indicate that Special Cycle recognition is disabled. |
| Memory Write and Invalidate Enable | 4 | 0b | Memory Write and Invalidate Enable. |
| VGA Palette Snoop | 5 | 0b | Hardwired to 0 to indicate that the VGA Palette Snoop is disabled. The controller does not need to snoop VGA palette cycles. |
| Parity Error Response | 6 | 0b | PERR# (Response) Detection Enable bit |
| Stepping Control | 7 | 0b | Hardwired to 0 to indicate that the Wait Cycle is disenabled. The controller does not need to insert a wait state between the address and data on the AD lines. |
| SERR# Enable | 8 | 0b | SERR# enable |
| Fast Back-to-Back Enable | 9 | 0b | Hardwired to 0 to indicate that Fast Back-to-back is disabled. The controller only acts as a master to a single device, so this functionality is not needed. |
| INTA# Enable# | 10 | 0b | When it is 0, INTA# is allowed to send out. When it is 1, INTA# is not allowed to send out. |
| Reserved | 15:11 | 00h | |
| Command Register: The PCI specification defines this register to control a PCI device's ability to generate and respond to PCI cycles. | | | |

| STATUS- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 0h | |
| Interrupt A status | 3 | 0b | When there is interrupt A, this bit will be 1, regardless of the value in reg0x04[10]. |
| Capabilities List | 4 | 1b | Read only. Indicates that the new capabilities list pointer configuration register is implemented in reg0x34. |
| 66MHz-Capable | 5 | 1b | Read only. Indicates that device is capable of running at 66MHz. |
| Reserved | 6 | 0b | |
| Fast Back-to-Back Capable | 7 | 0b | Read only. Indicates that device does not support fast back-to-back transactions. |
| Master Data Parity Error | 8 | 0b | Master Data Parity Error. This bit is set to 1 when the controller detects master data parity error. Cleared by writing a 1 to it. |
| Device Select Timing | 10:9 | 10b | DEVSEL# timing – Read only bits indicating DEVSEL# timing when performing a positive decode. |
| Signaled Target Abort | 11 | 0b | Read only. The device does not support target aborts. |
| Received Target Abort | 12 | 0b | Received Target Abort .This bit is set to 1 when the controller, acting as a PCI master, is aborted by a PCI target. Cleared by writing a 1 to it. |
| Received Master Abort | 13 | 0b | Received Master Abort Status. Set to 1 when the controller, acting as a PCI master, aborts a PCI bus memory cycle. Cleared by writing a 1 to it. |
| Signaled System Error | 14 | 0b | SERR# status. This bit is set to 1 when the controller detects a PCI address parity error. Cleared by writing a 1 to it. |
| Detected Parity Error | 15 | 0b | Detected Parity Error. This bit is set to 1 when the controller detects a parity error. Cleared by writing 1 to it. |
| Status Register: The PCI specification defines this register to record status information for PCI related events. This is a read/write register. However, writes can only reset bits. | | | |

| Revision ID/Class Code - R - 32 bits - [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | Revision ID. |
| Class Code | 31:8 | 040100h | Class Code. |
| Revision ID/Class Code Register: This read only register contains the device's revision information and generic function. | | | |

| Cache Line Size - RW - 8 bits - [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | Cache Line Size. |
| Cache Line Size Register: This register specifies the system cache line size. | | | |

| Latency Timer - RW - 8 bits - [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer | 7:0 | 00h | Latency Timer. |
| Latency Timer Register: This register specifies the value of the Latency Timer in units of PCICLKs. | | | |

| Header Type - R - 8 bits - [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 80h | Header Type. |
| Header Type Register: This register identifies the type of the predefined header in the configuration space. Since SB600 is a multifunction device, the most significant bit is set. | | | |

| BIST- R – 8 bits – [PCI_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST | 7:0 | 00h | BIST register. |
| Built-in Self Test Register: This register is used for control and status for Built-in Self Test. Ac97 has no BIST modes. | | | |

| Base Address Reg 0- RW - 32 bits – [PCI_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MemoryIndicator | 0 | 0b | Always 0; meaning it is always memory mapped |
| Type | 2:1 | 00b | Always 0; meaning it can be located anywhere in 32 bit address space |
| Prefetchable | 3 | 0b | Always 0; meaning it is not prefetchable |
| Reserved | 7:4 | 0h | Always 0; meaning the memory mapped registers occupy 256 bytes |
| BAR0 | 31:8 | 0000_00h | Base address register 0. Defines the base address for the memory mapped register space of audio. If index 50h, bit 3 is set, bits [13:8] of this register become unwritable. The effect will cause the OS to allocate a wider memory map for this controller. |

| Base Address Reg 1- RW - 32 bits – [PCI_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MemoryIndicator | 0 | 0b | This bit will return 1 if index 50h, bit [1] is set. 1 means IO mapping |
| Type | 2:1 | 00b | Always 0; meaning that it can be located anywhere in 32 bit address space. |
| Prefetchable | 3 | 0b | Always 0; meaning that it is not prefetchable. |

| Base Address Reg 1- RW - 32 bits – [PCI_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:4 | 0h | Always 0; meaning that the IO mapped registers occupy 256 bytes. |
| BAR1 | 31:8 | 0000_00h | Base address register 1. Defines the base address for the IO mapped register space of audio. This is to allow dual IO and memory mapped addressing. In other words, if IO mapping is enabled through this register, the IO indexed registers are basically dual port as the memory mapped registers. This register can only be programmed if index 50h, bit [1] is set. The purpose of this dual address mapping is for diagnostics; not for OS usage |

| Subsystem ID & Subsystem Vendor ID – W/R – 32 bits – [PCI_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Subsystem Vendor ID. |
| Subsystem ID | 31:16 | 0000h | Subsystem ID. |
| This 4-byte register is a write-once & read-only afterward register. The BIOS writes to this register once (all 4 bytes at once) & software reads its value when needed. | | | |

| Capabilities Pointer – R – 8 bits – [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 40h | Read only. Indicates that the device has New Capabilities register set starting at address 40h. |

| Interrupt Line – RW – 8 bits – [PCI_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | Identifies which input of the system interrupt controller the function's PCI interrupt request pin (as specified in its Interrupt Pin register) is routed to. The south bridge itself does not use this value, rather it is used by device drivers and operating systems. |

| Interrupt Pin – R – 8 bits – [PCI_Reg: 3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 02h | Hard-wired to 2 to indicate that this function (audio controller) uses interrupt pin INTB# on PCI bus.. |

| Min_Gnt – R – 8 bits – [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Min_Gnt | 7:0 | 02h | Hardwired to 2 to indicate the bus master would like to retain PCI bus ownership for 500ns during a cycle. |

| Max_Lat – R – 8 bits – [PCI_Reg: 3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Max_Lat | 7:0 | 00h | Hard-wired to 0 to indicate the bus master has no stringent requirement as to how often the device needs access to the PCI bus. |

| MSI Capability Register Set IDs- R – 16 bits – [PCI_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 05h | Read only. 05h indicates it is an MSI capability register set. |
| Pointer to Next ID | 15:8 | 00h | Read only. 00h indicates there is no additional register set. |

| MSI Message Control Register- RW – 16 bits – [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Enable | 0 | 0b | 0 - Function Is disabled from using MSI.<br>1 - Function is enabled to use MSI. |
| Multiple Message Capable | 3:1 | 0h | Hardwired to 0 to indicate the device would like 1 message allocated to it. |
| Multiple Message Enable | 6:4 | 0h | Read/Write. Software programs a 3-bit value into this field indicating the actual number of messages allocated to the device. The number allocated can be equal or less than the number actually requested. The field is encoded as follows:<br>Value Number of Messages Requested<br>000b 1<br>001b 2<br>010b 4<br>011b 8<br>100b 16<br>101b 32<br>110b reserved<br>111b reserved |
| 64-bit Address Capable | 7 | 0b | Hardwired to 0 to indicates that function does not implement the upper 32 bits of the Message Address register and is incapable of generating a 64-bit memory address. |
| Reserved | 15:8 | 00h | |

| MSI Message Address Register- RW – 32 bits – [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | | Reserved. |
| MSI Address | 31:2 | 0000_0000h | Lower 32 bits of the system specified message address always DW aligned. |

| MSI Message Data Register- RW – 16 bits – [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Data | 15:0 | 0h | System-specified message. |

| MSI Program Weight- RW – 8 bits – [PCI_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Program Weight | 5:0 | 04h | This register specifies the programmable priority of audio device's message signaled interrupt request. |
| Reserved | 7:6 | 0h | |

| UnMask Latency Timer Expiration W - 32 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| UnMask Latency Timer Expiration | 0 | 0b | When this bit is set to 0, latency timer register will be ignored, and AC97's PCI master will not time out.<br>this bit is write-only, i.e., reading from it always returns 0. |
| Base1Enable | 1 | 0b | When set, Base 1 (offset 14h) becomes writeable. |
| ReqMask | 2 | 0b | This is for internal bus performance enhancement. |
| LargeMemEnable | 3 | 0b | When set, bits [13:8] of base 0 (offset 10h) becomes unwritable.<br>This is to cause OS to allocate wider memory map for ac97. |
| Reserved | 31:4 | 0000_0000h | |

#### *2.5.1.2 Audio Memory Mapped Registers*

All AC'97 controller audio registers are mapped to the memory.

| Name | Reg. |
|---|---|
| Interrupt | 00h |
| Interrupt Enable | 04h |
| Audio Command | 08h |
| Output Phy Status and Address | 0Ch |
| Input Phy Address & Data | 10h |
| SLOTREQ | 14h |
| Counter | 18h |
| Input FIFO Threshold | 1ch |
| Input DMA Link List Pointer | 20h |
| Input DMA DT Start | 24h |
| Input DMA DT Next | 28h |
| Input DMA DT Current | 2Ch |
| Input DT Size And FIFO Info | 30h |
| Out DMA Slot Enbl. & Thresh | 34h |
| Out DMA Link List Pointer | 38h |
| Out DMA DT Start | 3Ch |
| Out DMA DT Next | 40h |
| Out DMA DT Current | 44h |
| Out DMA DT Size and State | 48h |
| SPDIF Command | 4ch |
| SPDIF Link List Pointer | 50h |
| SPDIF DT Start | 54h |
| SPDIF DT Next | 58h |
| SPDIF DT Current | 5Ch |
| SPDIF DT Size & FIFO Info | 60h |
| Modem Mirror | 7Ch |
| Audio Mirror | 80h |
| 6-Channel Reorder Enable | 84h |
| Audio FIFO Flush | 88h |
| Output DMA FIFO Info | 8Ch |
| SPDIF status bits reg1 | 90h |
| SPDIF status bits reg2 | 94h |
| SPDIF status bits reg3 | 98h |
| SPDIF status bits reg4 | 9Ch |
| SPDIF status bits reg5 | A0h |
| SPDIF status bits reg6 | A4h |
| Audio Phy Semaphore | A8h |

| Interrupt - RW - 32 bits - [MEM_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA Overflow | 0 | 0b | Input Channel overflow on the next AC'97 clock - out of FIFO space. |
| in DMA Status | 1 | 0b | Set to “1” after finishing an input audio DT data block (if reg0x04[1]=1 and reg0x08[3]=0). |
| out DMA Underflow | 2 | 0b | Output Channel underflow on the next AC'97 clock – no data in FIFO |
| out DMA Status | 3 | 0b | Set to "1" after finishing an output audio DT data block (if reg0x04[1]=1 and reg0x08[3]=0). |
| SPDIF Underflow | 4 | 0b | SPDIF is out of data |
| SPDIF Status | 5 | 0b | SPDIF status bit - set to "1" after finishing an SPDIF DT data block (if reg0x04[5]=1 and reg0x08[5]=0). |
| Reserved | 7:6 | 0b | |
| Phy Data Incoming | 8 | 0b | Got OR’ed Physical register address and data from Codecs |
| Phy Addr Mismatch | 9 | 0b | There is mismatch between in Physical and out Physical address values |
| Codec0 Not Ready | 10 | 0b | The Ac97_Phy registers in the master Ac97 codec are not ready for normal operation |
| Codec1 Not Ready | 11 | 0b | The Ac97_Phy registers in the 1st slave Ac97 codec are not ready for normal operation |
| Codec2 Not Ready | 12 | 0b | The Ac97_Phy registers in the 2nd slave Ac97 codec are not ready for normal operation |
| New Frame Starts | 13 | 0b | This bit is set when new frame starts |
| Reserved | 14 | 0b | |
| Audio Gpio Interrupt | 15 | 0b | When the input audio GPIO interrupt is enabled, input bus slot 12 bit 0 is considered as audio GPIO data. When that is true, if slot 12 is valid and bit 0 changes, this bit is set to indicate audio GPIO interrupt. |
| Reserved | 31:16 | 0000h | |
| Interrupt Source Register: Each bit in this register expresses an error flag. "1" indicates the error. Driver can read status or clear by writing “1”. Writing 0 to bit doesn't change its value. | | | |

| Interrupt Enable- RW - 32 bits - [MEM_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA Overflow en. | 0 | 0b | Enable Input Channel overflow interrupt. |
| Audio Status Enable | 1 | 0b | 1- When an input or output audio DT data block is finished, status will be updated in either DT memory or in reg0x00 (depending on reg0x08[3]).<br>0 – Don’t update status |
| out DMA Underflow en | 2 | 0b | Enable Output channel 0 underflow interrupt. |
| Out DMA Underflow Condition Select | 3 | 0b | 0—Underflow interrupt is asserted only when output DMA FIFO has zero valid entry.<br>1—Underflow interrupt is asserted as long as output DMA FIFO does not have enough valid entries for the coming frame |
| SPDIF Underflow en | 4 | 0b | Enable SPDIF underflow interrupt. |
| SPDIF Status enable | 5 | 0b | 1- When an SPDIF DT data block is finished, status will be updated in either DT memory or in reg0x00 (depending on reg0x08[5]).<br>0 – Don’t update status |
| Reserved | 7:6 | 00b | |
| Phy in Interrupt en | 8 | 0b | Enable "Got Physical register data from Codec" interrupt |
| Phy_addr_mismatch_en | 9 | 0b | Enable Physical address in/out mismatch interrupt |
| Codec0 Not Ready En | 10 | 0b | Enable Codec0_not_ready interrupt |
| Codec1 Not Ready En | 11 | 0b | Enable Codec1_not_ready interrupt |
| Codec2 Not Ready En | 12 | 0b | Enable Codec2_not_ready interrupt |
| New Frame Start En | 13 | 0b | Enable new frame start interrupt |
| Set Bus Busy Audio | 14 | 0b | Audio is running (write only). Set/cleared by software. |
| Audio gpio interrupt en | 15 | 0b | Enable audio GPIO interrupt |

| Interrupt Enable- RW - 32 bits - [MEM_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 31:16 | 0000h | |
| Interrupt Enable Register: If a bit in this register is set to “1”, the corresponding interrupt is enabled. Default—all disabled. | | | |

| Audio Command- RW - 32 bits - [MEM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Power down | 0 | 0b | When this bit is set, the INTA# will be only set when any of the SDATA_IN is '1'. Should be used for catching of PME# event. |
| Audio receive enable | 1 | 0b | Enable receiving of audio data from AC-link. |
| Audio send out enable | 2 | 0b | Enables sending of audio data to AC97 link. |
| Audio Status To Mem | 3 | 0b | When reg0x04[1] is enabled, after finishing an input or output audio DT data block, the status will be updated in either DT memory or in reg0x00.<br>0 – Update status to register.<br>1 – Update status to DT memory. |
| SPDIF send out en | 4 | 0b | Enable SPDIF to send out data. Also it enables Underflow interrupt for SPDIF. |
| SPDIF Status To Mem | 5 | 0b | When reg0x04[5] is enabled, after finishing an SPDIF DT data block, the status will be updated in either DT memory or in reg0x00.<br>0 – Update status to register.<br>1 – Update status to DT memory. |
| Modem_slots_alloc_1 | 6 | 0b | Mirror bit of modem memory mapped register0x08[6]. This is only available in ASIC revision A32 or later. Refer to that bit for more detailed information. |
| Reserved | 7 | 0b | |
| in DMA enable | 8 | 0b | Enable input DMA |
| out DMA enable | 9 | 0b | Enable output DMA |
| SPDIF DMA enable | 10 | 0b | Enable SPDIF DMA |
| SPDIF out Stopped | 11 | 0b | 1 - the SPDIF_OUT is set to the 0 because SPDIF FIFO run out of data or SPDIF_en =0<br>0 - the SPDIF_OUT is processing the active data. [Read-only] |
| SPDIF concurrency | 14:12 | 0h | 000 – no SPDIF concurrency, SPDIF data go to SPDIF bus.<br>001 – SPDIF data go to slot 3 & 4<br>010 – SPDIF data go to slot 7 & 8<br>011 – SPDIF data go to slot 6 & 9<br>100 – SPDIF data go to slot 10 & 11 |
| Reserved | 15 | 0b | |
| SPDIF interleave enable | 16 | 0b | 0 – Disable SPDIF interleave mode<br>1 – Enable SPDIF interleave mode. In this mode a 32-bit PCI dword will split into two 16-bit words and be put onto SPDIF bus as two frames. |
| Reserved | 19:17 | 0h | |
| Audio present | 20 | 0b | 1 - Audio is present<br>0 - Audio is not present (for Modem driver), set by BIOS [Read-only] |
| Interleave in | 21 | 0b | 1 - Enable Interleave format for the input slot 3 and 4 data, 31:16 - slot 4, 15:0- slot 3.<br>Also this disables catching and transferring of slot 6 data.<br>0 - Enable transfer of slot 3,4 and 6 in regular format, where bits 31:12 - data and bits 11:8 are slot id (11 is slot 3, 100 is slot 4 and 110 is slot 6) |
| Interleave out | 22 | 0b | 1 - Enable transferring from memory the data in interleave format - the upper part of DWORD [31:16] is right data and lower part of DWORD [15:0] is left data.<br>0 - Data from memory is transferred in regular way - one DWORD , one data. |
| Loop back enable | 23 | 0b | Enable loop-back mode. The SDATA_OUT connected directly to the SDATA_IN |

| Audio Command- RW - 32 bits - [MEM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Packed format disable | 24 | 0b | 1 - Disable packed format for sending data: data is sent to memory including data from Slot1 and 2 (physical address and data are send to memory along with other data).<br>0 - Data send to memory through DMA in packed format, no slot 1 and 2 - default |
| Burst enable. | 25 | 0b | Enable DMA operation in burst mode: Will send/request data only based on threshold value & overflow condition. All other cases (lowest priority) will be disabled. |
| Panic enable | 26 | 0b | Enable panic signal - to get more priority over internal bus. |
| Modem_slots_alloc_0 | 27 | 0b | Mirror bit of modem memory mapped register0x08[27]. Refer to that bit for more detailed information. |
| AC Link active | 28 | 0b | Read only.<br>0 - AC Link is not active;<br>1 - AC Link is active. |
| AC'97 software reset | 29 | 0b | 0 – De-assert ac97 software reset asynchronous to BIT_CLK;<br>1 – Assert ac97 software reset asynchronous to BIT_CLK. Registers in BIT_CLK domain will be reset. |
| AC97 Sync | 30 | 0b | 0 – De-assert AC link's SYNC asynchronous to BIT_CLK<br>1 – Assert AC link's SYNC asynchronous to BIT_CLK. After being asserted, the bit will automatically deassert itself in one clock. |
| AC97 Reset# | 31 | 0b | 0 – Assert AC link's RESET# asynchronous to BIT_CLK;<br>1 – De-assert AC link's RESET# asynchronous to BIT_CLK |
| Audio Command Register: Controls the operation of Audio Controller. Value of "1" in bit position enables corresponding function. Value of "0" disables it. | | | |

| Phy Out Address and Data- RW - 32 bits - [MEM_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Codec ID | 1:0 | 0b | 0 – Master AC97<br>1 – 1st slave AC97<br>2 – 2nd slave AC97<br>3 – Reserved |
| Read/Write request | 2 | 0b | 1 – Read request.<br>0 – Write request |
| Reserved | 7:3 | 0h | |
| Phy out enable | 8 | 0b | 1 - Enable sending out of Physical address in next frame<br>0 - Physical out address have been sent out; open to get new value. When the bit is 0, writing 1 to it makes it 1. Once it is 1, writing to it is ignored. It will automatically return to 0 when physical address is sent out. |
| Phy out address | 15:9 | 00h | Physical out address (will not be able to write new address until old one is send out - until Physical out address enable is '1') (for address write bit [8] is asserted until data send out) |
| Phy out Data | 31:16 | 0000h | Physical register out data (for write). When Physical out address enabled and Write request is set (bit [2]=0), then the value in this register will be sent out in slot 2 |
| Phy out address and data Register: Until Physical out address enable is 0, driver cannot write any new data to this register. | | | |

| Input Phy Address and Data- R - 32 bits - [MEM_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |
| Phy_in_read_flag | 8 | 0h | It is cleared whenever reg0x0Ch[8]=1 and reg0x0C[2]=1.<br>It is set when input physical data (due to a physical read) arrives. |
| in_Phy_addr | 15:9 | 00h | Physical address from AC'97 Codec. |
| Input Phy data | 31:16 | 0000h | Input Physical data from AC'97 Codec. |
| Input Phy address and data Register | | | |

| Slot Request- R - 32 bits - [MEM_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SLOTREQ | 9:0 | 000h | The read only bits [0:9] of this field respectively come from slot1[11:2] of AC link's SDATA_IN0/1/2 OR'ed together (even if slot 1 may not be valid as indicated by slot 0[14]). These bits [0:9] respectively decide whether slot 3~12 of SDATA_OUT is allowed or not.<br>0 – Slot is allowed<br>1 – Slot is allowed<br>Even if a bit in this field being 0 indicates a slot is allowed, the controller further looks at reg0x34[9:0] to finally decide whether the slot is enabled. |
| Reserved | 31:10 | 000000h | |
| SLOTREQ Register | | | |

| Counter- R - 32 bits - [MEM_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Slot Counter | 3:0 | 0h | The current slot number (0-12) which the AC97 controller handling. |
| Reserved | 7:4 | 0h | |
| Bit Clock Counter | 12:8 | 00h | For tag slot, the value changes from 0 to F. For other slots the value changes from 0 to 20. |
| Reserved | 31:13 | 00000h | |
| Counter Register | | | |

| Input FIFO Threshold- RW - 32 bits - [MEM_Reg: 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input Threshold | 4:0 | 00h | Threshold value for the Input Channel's FIFOs. (FIFO size 16x20) |
| Reserved | 31:5 | 0000000h | |
| Input FIFO Threshold Register: Input DMA's FIFOs threshold value. | | | |

| Input DMA Link List Pointer- RW - 32 bits - [MEM_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA Link List Pointer En | 0 | 0b | Input DMA Link List Pointer enable |
| in DMA Link List Pointer | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Input DMA Link List Pointer Register: | | | |

| Input DMA DT Start- R - 32 bits - [MEM_Reg: 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the input DMA |
| Input DMA Discrete Table (DT) Start Pointer Register: | | | |

| Input DMA DT Next- R - 32 bits - [MEM_Reg: 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT next | 31:0 | 0000_0000h | Pointer to the next DT for the input DMA |
| Input DMA DT Next Pointer Register: | | | |

| Input DMA DT Current- R - 32 bits - [MEM_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the input DMA. |
| Input DMA DT Current Pointer Register: | | | |

| Input DMA DT Size & FIFO Info- R - 32 bits - [MEM_Reg: 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT size | 15:0 | 0000h | Data size of DT for input DMA. |
| in FIFO Used | 20:16 | 00h | Number of filled FIFO entries of input DMA. |
| in FIFO Free | 25:21 | 1eh | Number of free FIFO entries of input DMA. |
| in DMA state | 28:26 | 0h | Current state of in DMA |
| Reserved | 31:29 | 0h | |
| Input DMA DT Size & FIFO info Register: Size of data associated with DT for input channels and number of FIFO entries free and used for the input channels. | | | |

| Output DMA Slot Enable & Threshold- RW - 32 bits - [MEM_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA Slot enable | 9:0 | 000h | Slot enable for audio output.<br>[0] = 1 Enable slot 3<br>[1] = 1 Enable slot 4<br>\|<br>[9] = 1 Enable slot 12 |
| Reserved | 10 | 0b | |
| Out DMA Threshold | 17:11 | 00h | Threshold value for the Out DMA FIFO. (FIFO size 90x20) |
| Reserved | 31:18 | 0000h | |
| Output DMA Slot Enable & Threshold Register: Configure Slot responsibility for Output channels 0 and also threshold values of out DMA FIFO's. | | | |

| Output DMA Link List Pointer- RW - 32 bits - [MEM_Reg: 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA Link List Pointer En | 0 | 0b | Output DMA Link List Pointer enable |
| out DMA Link List Pointer | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Output DMA Link List Pointer Register: | | | |

| Output DMA DT Start- R - 32 bits - [MEM_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the Output DMA. |
| Output DMA Discrete Table (DT) Start Pointer Register: | | | |

| Output DMA DT Next- R - 32 bits - [MEM_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA DT next | 31:0 | 0000_0000h | Pointer to the next DT for the Output DMA |
| Output DMA DT Next Pointer Register: | | | |

| Output DMA DT Current- R - 32 bits - [MEM_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the Output DMA. |
| Output DMA DT Current Pointer Register: | | | |

| Output DMA DT Size and State- R - 32 bits - [MEM_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Out DMA DT size | 15:0 | 0000h | Data size of DT for Output DMA. |
| Reserved | 25:16 | 000h | |
| out DMA state | 28:26 | 0h | Current state of out DMA |
| Reserved | 31:29 | 0h | |
| Output DMA DT Size & State Register: | | | |

| SPDIF Command- RW - 32 bits - [MEM_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Spdif valid bit | 0 | 0b | The value of this bit will go to bit[28] of each enabled sub-frame on SPDIF_OUT bus, which is the valid bit for that subframe. |
| Spdif user bit | 1 | 0b | The value of this bit will go to bit[29] of each enabled sub-frame on SPDIF_OUT bus, which is the user bit for that subframe. |
| SPDIF bus mute | 2 | 0b | Setting it to 1 will mute the whole SPDIF bus to zero when the next frame comes. Setting it back to 0 will unmute SPDIF bus when the next frame comes. Changing of this bit only starts to take effect on frame boundary. |
| Reserved | 3 | 0b | |
| LFSR Control | 4 | 0b | 0 - Shift direction left.<br>1 - Shift direction right. |
| Single channel mode | 5 | 0b | 1 - SPDIF operates in the single channel mode (disable second sub-frame)<br>0 - SPDIF operates in the 2-channel mode (every sub-frame filed with data from FIFO) |
| Reserved | 6 | 0b | |
| Data mute | 7 | 0b | When this bit is set to 1, on each subframe of spdif bus, the data field (bit[27:4]) is set to 0, no matter what data are written to spdif FIFO. |
| LFSR Accumulation | 15:8 | 00h | LFSR Linear feedback shift Register [Read-only] |
| SPDIF Threshold | 20:16 | 00h | Threshold value for the SPIDF Channel's FIFO. (SPDIF FIFO is 30x28) |
| Reserved | 31:21 | 000h | |
| SPDIF Command Register: Controls the operation of SPDIF channel. | | | |

| SPDIF Channel Link List Pointer- RW - 32 bits - [MEM_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF LL ptr enable | 0 | 0b | SPDIF Channel Link List Pointer enable |
| SPDIF LL Pointer | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| SPDIF Channel Link List Pointer Register: Pointer to the start of Link List for the SPDIF channel. | | | |

| SPDIF Channel DT Start- R - 32 bits - [MEM_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF DT start | 31:0 | 0000_0000h | SPDIF Channel Discrete Table (DT) Start Pointer |
| SPDIF channel DT start Pointer Register: Pointer to the start of data associated with current DT for the SPDIF channel. | | | |

| SPDIF Channel DT Next- R - 32 bits - [MEM_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF DT next | 31:0 | 0000_0000h | Pointer to the next DT for the SPDIF channel. |
| SPDIF channel Next Pointer Register: | | | |

| SPDIF Channel DT Current- R - 32 bits - [MEM_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF DT current | 31:0 | 0000_0000h | SPDIF Channel currently accessed memory address. |
| SPDIF DT Current Pointer Register: Pointer to the currently accessing memory address for the SPDIF channel. | | | |

| SPDIF DT Size- R - 32 bits - [MEM_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF DT size | 15:0 | 0000h | Data size of DT for SPDIF channel. |
| SPDIF Used | 20:16 | 00h | Number of filled FIFO entries of SPDIF channel. Default is 0. |
| SPDIF Free | 25:21 | 1Eh | Number of free FIFO entries of SPDIF channel. Default is 30 (Size of FIFO) |
| SPDIF DMA state | 28:26 | 0h | Current state of SPDIF DMA |
| Reserved | 31:29 | 0h | |
| SPDIF DT Size and FIFO Used/Free Register: Size of data associated with DT for SPDIF channels. Plus Number of free and filled FIFO entries of SPDIF channel. | | | |

| Modem Mirror- R - 32 bits - [MEM_Reg: 7Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Modem Mirror | 31:0 | 0000_0000h | Data written by modem for communication with Audio. |
| Modem Mirror Register: | | | |

| Audio Mirror- RW - 32 bits - [MEM_Reg: 80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Audio Mirror | 31:0 | 0000_0000h | Data written by audio for communication with modem. Bit 1 is used by the BIOS to communicate with the audio driver. |
| Audio mirror Register: | | | |

| 6-Channel Reorder Enable- RW - 32 bits - [MEM_Reg: 84h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reorder enable | 0 | 0 | When audio output has all the 6 channels enabled, usually software puts data in the memory using the order of 346789, and AC97 controller sends data out with the same order 346789. If software uses the order of 346978, with this bit enabled, AC97 controller can still send data out with the correct order 346789. |
| Reserved | 31:1 | 0000_0000h | |
| 6-Channel Reorder Enable Register | | | |

| Audio Fifo Flush- W - 32 bits - [MEM_Reg: 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Flush output audio fifo | 0 | 0b | Writing to this bit will flush audio output DMA fifo, i.e., the fifo indexes and Used/Free counts will be reset. Reading this bit returns 0 |
| Flush input audio fifo | 1 | 0b | Writing to this bit will flush audio input DMA fifo, i.e., the fifo indexes and Used/Free counts will be reset. Reading this bit returns 0 |
| Flush spdif fifo | 2 | 0b | Writing to this bit will flush spdif output DMA fifo, i.e., the fifo indexes and Used/Free counts will be reset. Reading this bit returns 0 |
| Reserved | 31:3 | 0000_0000h | |
| Audio Fifo Flush Register: | | | |

| Output DMA Fifo info- R - 32 bits - [MEM_Reg: 8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Out FIFO Used | 6:0 | 00h | Number of filled FIFO entries of output DMA. |
| Reserved | 7 | 0b | |
| Out FIFO Free | 14:8 | 5Ah | Number of free FIFO entries of output DMA. Default is 90 (size of output fifo) |
| Reserved | 31:15 | 00000h | |

| SPDIF Status bits reg1- RW - 32 bits - [MEM_Reg: 90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | The bits in this reg are used on spdif bus as status bits. Each spdif bus frame has a status bit. Spdif bus frame counts from 0~191 repeatedly, so there are 192 status bits. Software can write to this register (and the following few registers) to decide what status bits value to output on spdif bus.<br>Bits 0~31 in this register correspond to frame 0~31 on spdif bus. |

| SPDIF Status bits reg2- RW - 32 bits - [MEM_Reg: 94h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | Same definition as reg0x90.<br>Bits 0~31 in this register correspond to frame 32~63 on spdif bus. |

| SPDIF Status bits reg3- RW - 32 bits - [MEM_Reg: 98h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | Same definition as reg0x90.<br>Bits 0~31 in this register correspond to frame 64~95 on spdif bus. |

| SPDIF Status bits reg4- RW - 32 bits - [MEM_Reg: 9Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | Same definition as reg0x90.<br>Bits 0~31 in this register correspond to frame 96~127 on spdif bus. |

| SPDIF Status bits reg5- RW - 32 bits - [MEM_Reg: A0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | Same definition as reg0x90.<br>Bits 0~31 in this register correspond to frame 128~159 on spdif bus. |

| SPDIF Status bits reg6- RW - 32 bits - [MEM_Reg: A4h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPDIF Status bits | 31:0 | 0000_0000h | Same definition as reg0x90.<br>Bits 0~31 in this register correspond to frame 160~191 on spdif bus. |

| Audio Phy Semaphore Reg- RW - 8 bits - [MEM_Reg: A8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Audio Phy semaphore | 0 | 0b | PHY is ready for Audio to access:<br>0 = PHY is not ready for Audio to access.<br>1 = PHY is ready for Audio to access.<br><br>The Audio driver can only access the PHY register when the read back value of this bit is '1' (after writing a '1' to this bit). It is possible that the read back value is '0' after writing a '1'. This means that the PHY register is not ready for Audio to access. Since the PHY register can be accessed by both audio and modem controllers, this bit is used by the audio driver to establish a semaphore handshake with the modem driver.<br><br>The driver is also responsible to clear the bit after its PHY read/write access.<br><br>Note: The driver can check Mem_reg 0x10[8] which indicates a PHY read access completion. For writing to PHY, the driver can clear the semaphore bit after the write command. |
| Reserved | 7:1 | 00h | |

### *2.5.2 Modem Registers (Device 20, Function 6)*

The PCI based registers for Modem are defined according to the PCI 2.1 spec and Windows 2000 requirements.

#### *2.5.2.1 PCI Configuration Registers*

| Register Name | Offset Address |
|---|---|
| Vendor ID | 00h |
| Device ID | 02h |
| PCI Command | 04h |
| PCI Device Status | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| Built-in Self Test | 0Fh |
| Base Address Register 0 | 10h |
| Base Address Register 1 | 14h |
| Subsystem ID & Subsystem Vendor ID | 2Ch |
| Capabilities Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Min Grant | 3Eh |
| Max Latency | 3Fh |
| MSI Capability Register Set IDs | 40h |
| MSI Message Control | 42h |
| MSI Message Address | 44h |
| MSI Message Data | 48h |
| MSI Program Weight | 4Ch |

**IMPORTANT: The driver is required to check revision ID to enable functions appropriately. AMD AC'97 controller will be backward compatible to previous revisions. E.g., Revision 01 will have everything the same as revision 00, plus the enhancement**.

| Vendor ID- R - 16 bits - [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | Vendor ID |
| Vendor ID Register: This register holds a unique 16-bit value assigned to a vendor, and combined with the device ID it identifies any PCI device. | | | |

| Device ID- R - 16 bits - [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID | 15:0 | 438Eh | Device ID |
| Device ID Register: This register holds a unique 16-bit value assigned to a device, and combined with the vendor ID it identifies any PCI device. | | | |

| CMD- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Space | 0 | 0b | I/O Access Enable. |
| Memory Space | 1 | 0b | Memory Access Enable. |
| Bus Master | 2 | 0b | Master Enable |
| Special Cycles | 3 | 0b | Hardwired to 0 to indicate that Special Cycle recognition is disabled. |
| Memory Write and Invalidate Enable | 4 | 0b | Memory Write and Invalidate Enable. |
| VGA Palette Snoop | 5 | 0b | Hardwired to 0 to indicate that the controller does not need to snoop VGA palette cycles. |
| Parity Error Response | 6 | 0b | PERR# (Response) Detection Enable bit |
| Stepping Control | 7 | 0b | Hardwired to 0 to indicate that the controller does not need to insert a wait state between the address and data on the AD lines. |
| SERR# Enable | 8 | 0b | SERR# enable |
| Fast Back-to-Back Enable | 9 | 0b | Hardwired to 0 to indicate that Fast Back-to-back is disabled. The controller only acts as a master to a single device, so this functionality is not needed. |
| INTB# Enable# | 10 | 0b | When it is 0, INTB# is allowed to send out. When it is 1, INTB# is not allowed to send out. |
| Reserved | 15:11 | 00h | |
| Command Register: The PCI specification defines this register to control a PCI device's ability to generate and respond to PCI cycles. | | | |

| STATUS- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 0h | |
| Interrupt B status | 3 | 0b | When there is interrupt B, this bit will be 1, regardless of the value in reg0x04[10]. |
| Capabilities List | 4 | 1b | Read only. Indicates that the new capabilities list pointer configuration register is implemented in reg0x34. |
| 66MHz-Capable | 5 | 1b | Read only. Indicates that device is capable of running at 66MHz. |
| Reserved | 6 | 0b | |
| Fast Back-to-Back Capable | 7 | 0b | Read only. Indicates that device does not support fast back-to-back transactions. |
| Master Data Parity Error | 8 | | Master Data Parity Error. This bit is set to 1 when the controller detects master data parity error. Cleared by writing a 1 to it. |
| Device Select Timing | 10:9 | 10b | DEVSEL# timing – Read only bits indicating DEVSEL# timing when performing a positive decode. |
| Signaled Target Abort | 11 | 0b | Read only. The device does not support target aborts. |
| Received Target Abort | 12 | 0b | Received Target Abort .This bit is set to 1 when the controller generated PCI cycle is aborted by a PCI target. Cleared by writing a 1 to it. |
| Received Master Abort | 13 | 0b | Received Master Abort Status. Set to 1 when the controller acting as a PCI master, aborts a PCI bus memory cycle. Cleared by writing a 1 to this bit. Cleared by writing a 1 to it. |
| Signaled System Error | 14 | 0b | SERR# status. This bit is set to 1, when the controller detects a PCI address parity error. Cleared by writing a 1 to it. |
| Detected Parity Error | 15 | 0b | Detected Parity Error. This bit is set to 1 when the controller detects a parity error. |
| Status Register: The PCI specification defines this register to record status information for PCI related events. This is a read/write register. However, writes can only reset bits. | | | |

| Revision ID/Class Code – R - 32 bits - [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | |
| Class Code | 31:8 | 070300h | Class Code. |
| Revision ID Register: This read only register contains the device's revision information and generic function. | | | |

| Cache Line Size - RW - 8 bits - [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | Cache Lien Size. |
| Cache Line Size Register: This register specifies the system cache line size. | | | |

| Latency Timer - RW - 8 bits - [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer | 7:0 | 00h | Latency Timer. |
| Latency Timer Register: This register specifies the value of the Latency Timer in units of PCICLKs. | | | |

| Header Type - R - 8 bits - [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 80h | Header Type. |
| Header Type Register: This register identifies the type of the predefined header in the configuration space. Since SB600 is a multifunction device, the most significant bit is set. | | | |

| BIST- R - 8 bits - [PCI_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST | 7:0 | 00h | BIST. |
| Built-in Self Test Register: This register is used for control and status for Built-in Self Test. Ac97 has no BIST modes. | | | |

| Base Address Reg 0- RW* - 32 bits – [PCI_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MemoryIndicator | 0 | 0b | Always 0; meaning it is always memory mapped |
| Type | 2:1 | 00b | Always 0; meaning it can be located anywhere in 32 bit address space |
| Prefetchable | 3 | 0b | Always 0; meaning it is not prefetchable |
| Reserved | 7:4 | 0h | Always 0; meaning the memory mapped registers occupy 256 bytes |
| BAR0 | 31:8 | 0000_00h | Base address register 0. Defines the base address for the memory mapped register space of modem. If index 50h, bit 3 is set, bits [13:8] of this register become unwritable. The effect will cause the OS to allocate a wider memory map for this controller. |

| Base Address Reg 1- RW* - 32 bits – [PCI_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MemoryIndicator | 0 | 0b | This bit will return 1 if index 50h, bit [1] is set. 1 means IO mapping. |
| Type | 2:1 | 00b | Always 0; meaning it can be located anywhere in 32 bit address space |
| Prefetchable | 3 | 0b | Always 0; meaning it is not prefetchable |
| Reserved | 7:4 | 0h | Always 0; meaning the IO mapped registers occupy 256 bytes |
| BAR1 | 31:8 | 0000_00h | Base address register 1. Defines the base address for the IO mapped register space of modem. This is to allow dual IO and memory mapped addressing. In other words, if IO mapping is enabled through this register, the IO indexed registers are basically dual port as the memory mapped registers. This register can only be programmed if index 50h, bit [1] is set. The purpose of this dual address mapping is for diagnostics; not for OS usage |

| Subsystem ID & Subsystem Vendor ID - W/R - 32 bits - [PCI_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Subsystem Vendor ID. |
| Subsystem ID | 31:16 | 0000h | Subsystem ID. |
| This 4-byte register is a write-once & read-only afterward register. The BIOS writes to this register once (all 4 bytes at once) & software reads its value when needed. | | | |

| Capabilities Pointer - R - 8 bits - [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 40h | Read only. Indicates that the device has New Capabilities register set starting at address 40h. |

| Interrupt Line - RW - 8 bits - [PCI_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | Identifies which input on the interrupt controller the function's PCI interrupt request pin (as specified in its Interrupt Pin register) is routed to. |

| Interrupt Pin - R - 8 bits - [PCI_Reg: 3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 02h | Hard-wired to 2 to indicate that this function (modem controller) uses interrupt pin INTB# on PCI bus. |

| Min_Gnt - R - 8 bits - [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Min_Gnt | 7:0 | 02h | The value hardwired into this register indicates how long the bus master would like to retain PCI bus ownership. |

| Min_Gnt - R - 8 bits - [PCI_Reg: 3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Max_Lat | 7:0 | 00h | This read-only register specifies how often the device needs access to the PCI bus (in increments of 250ns). |

| MSI Capability Register Set IDs- R – 16 bits - [PCI_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 05h | Read only. 05h indicates it is an MSI capability register set. |
| Pointer to Next ID | 15:8 | 00h | Read only. 00h indicates there is no additional register set. |

| MSI Message Control Register- RW - 16 bits - [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Enable | 0 | 0b | 0 - Function Is disabled from using MSI.<br>1 - Function is enabled to use MSI. |
| Multiple Message Capable | 3:1 | 0b | Hardwired to 0 to indicate the device would like 1 message allocated to it. |

| MSI Message Control Register- RW - 16 bits - [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Multiple Message Enable | 6:4 | 0b | Software programs a 3-bit value into this field indicating the actual number of messages allocated to the device. The number allocated can be equal or less than the number actually requested. The field is encoded as follows:<br>Value Number of Messages Requested<br>000b 1<br>001b 2<br>010b 4<br>011b 8<br>100b 16<br>101b 32<br>110b Reserved<br>111b Reserved |
| 64-bit Address Capable | 7 | 0b | Hardwired to 0 to indicates that function does not implement the upper 32 bits of the Message Address register and is incapable of generating a 64-bit memory address. |
| Reserved | 15:8 | 00h | |

| MSI Message Address Register- RW – 32 bits - [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | | Reserved. |
| MSI Address | 31:2 | 0000_0000h | Lower 32 bits of the system specified message address always DW aligned. |

| MSI Message Data Register- RW – 16 bits - [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Data | 15:0 | 0h | System-specified message. |

| MSI Program Weight Register RW- 8 bits - [PCI_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Program Weight | 5:0 | 000100b | This register specifies the programmable priority of modem device's message signaled interrupt request. |
| Reserved | 7:6 | 0h | |

| UnMask Latency Timer Expiration W - 32 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| Base1Enable | 1 | 0b | When set, Base 1 (offset 14h) becomes writeable. |
| Reserved | 2 | 0b | |
| LargeMemEnable | 3 | 0b | When set, bits [13:8] of base 0 (offset 10h) becomes unwritable. This is to cause OS to allocate wider memory map for ac97. |
| Reserved | 31:4 | 0000_000h | |

#### *2.5.2.2 Modem Memory Mapped Registers*

All the AC'97 Controller modem registers are mapped to memory.

| Register Name | Offset Address |
|---|---|
| Interrupt | 00h |
| Interrupt Enable | 04h |
| Modem Command | 08h |
| Output Phy Status And Address | 0Ch |
| Input Phy Address & Data | 10h |
| SLOTREQ | 14h |
| Counter | 18h |
| Input Fifo Threshold | 1Ch |
| Input DMA Link List Pointer | 20h |
| Input DMA DT Start | 24h |
| Input DMA DT Next | 28h |
| Input DMA DT Current | 2Ch |
| Input DT Size And FIFO Info | 30h |
| Out Fifo's Threshold | 34h |
| Output DMA1 Link List Pointer | 38h |
| Output DMA2 Link List Pointer | 3Ch |
| Output DMA3 Link List Pointer | 40h |
| Output DMA1 DT Start | 44h |
| Output DMA1 DT Next | 48h |
| Output DMA1 DT Current | 4Ch |
| Output DMA2 DT Start | 50h |
| Output DMA2 DT Next | 54h |
| Output DMA2 DT Current | 58h |
| Output DMA3 DT Start | 5Ch |
| Output DMA3 DT Next | 60h |
| Output DMA3 DT Current | 64h |
| Output DMA 1/2 Size | 68h |
| Output DMA 3 DT Size | 6Ch |
| Output Dma 1/2/3 Fifo Info | 70h |
| Output GPIO Out Data | 74h |
| Input GPIO Data | 78h |
| Modem Mirror | 7Ch |
| Audio Mirror | 80h |
| Modem FIFO Flush | 88h |
| Phy Semphore Register | A8h |

| Interrupt - RW - 32 bits - [MEM_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA Overflow | 0 | 0b | Input Channel overflow - out of FIFO space |
| in DMA Status | 1 | 0b | Input Channel status bit - set to "1" after finishing DT (if reg0x04[1]=1 and reg0x08[5]=1) |
| out DMA1 Underflow | 2 | 0b | Output DMA1 underflow - no data in FIFO |
| out DMA1 Status | 3 | 0b | Output DMA1 status bit - set to "1" after finishing DT (if reg0x04[1]=1 and reg0x08[5]=1) |
| out DMA2 Underflow | 4 | 0b | Output DMA2 underflow - no data in FIFO |
| out DMA2 Status | 5 | 0b | Output DMA2 status bit - set to "1" after finishing DT (if reg0x04[1]=1 and reg0x08[5]=1) |
| out DMA3 Underflow | 6 | 0b | Output DMA3 underflow - no data in FIFO |
| out DMA3 Status | 7 | 0b | Output DMA3 status bit - set to "1" after finishing DT (if reg0x04[1]=1 and reg0x08[5]=1) |
| Phy Data Incoming | 8 | 0b | Got OR'ed Phy register address and data from Codecs |

| Interrupt - RW - 32 bits - [MEM_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Phy Addr mismatch | 9 | 0b | There is mismatch between in Phy and out Phy address values |
| Codec0 Not Ready | 10 | 0b | The Ac97_Phy registers in the master Ac97 are not ready for normal operation |
| Codec1 Not Ready | 11 | 0b | The Ac97_Phy registers in the 1st slave Ac97 are not ready for normal operation |
| Codec2 Not Ready | 12 | 0b | The Ac97_Phy registers in the 2nd slave Ac97 are not ready for normal operation |
| New Frame Starts | 13 | 0b | Signal when new frame starts |
| Gpio in Data Intr. | 14 | 0b | Got Gpio data from Codec |
| Reserved | 31:15 | 00000h | |
| Interrupt Source Register: Each bit in this register expresses an error flag. "1" indicates the error. Driver can read status or clear by writing “1”. Writing 0 to bit doesn't change its value. | | | |

| Interrupt Enable - RW - 32 bits - [MEM_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA Overflow en | 0 | 0b | Enables Input Channel overflow interrupt. |
| Status enable | 1 | 0b | 1 – When an input or output modem DT data block is finished, status will be updated in either DT memory or in reg0x00 (depending on reg0x08[5]).<br>0 - Don’t update status |
| out DMA1 Underflow en | 2 | 0b | Enables out DMA3 underflow interrupt. |
| Reserved | 3 | 0b | |
| out DMA2 Underflow en | 4 | 0b | Enables out DMA3 underflow interrupt. |
| Reserved | 5 | 0b | |
| out DMA3 Underflow en | 6 | 0b | Enables out DMA3 underflow interrupt. |
| Reserved | 7 | 0b | |
| Phy in Intr. En | 8 | 0b | Enables "Got Physical register data from Codec" interrupt. |
| Phy_addr_mismatch_en | 9 | 0b | Enables Physical address in/out mismatch interrupt |
| Codec0_not_ready en | 10 | 0b | Enables Codec0 Not Ready interrupt |
| Codec1_not_ready en | 11 | 0b | Enables Codec1 Not Ready interrupt |
| Codec2_not_ready en | 12 | 0b | Enables Codec2_not_ready interrupt |
| New_Frame_start enable | 13 | 0b | Enables new frame start interrupt |
| Gpio in Data Intr. En | 14 | 0b | Enables "Got Gpio data from Codec" interrupt |
| Gpio in Data Intr. Diff En | 15 | 0b | This bit is effective only when “Gpio in Data Intr En” bit is asserted.<br>0 – Gpio interrupt happens whenever valid GPIO data come in<br>1 – Gpio interrupt happens only when valid GPIO data change in different frames. |
| Set Bus Busy Modem | 16 | 0b | Modem is running (write only). Set/cleared by software. |
| Reserved | 31:17 | 0000h | |
| Interrupt Enable Register: If a bit in this register is set to “1”, then the corresponding interrupt is enabled. Default -- all disabled. | | | |

| Modem Command - RW - 32 bits - [MEM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Power down | 0 | 0b | When this bit is set, the INTA# will be only set when any of the SDATA_IN is '1'. Should be used for catching of PME# event. |
| Modem receive enable | 1 | 0b | Enables receiving of modem data from AC link. |
| Modem send out through DMA1 enable | 2 | 0b | Enables sending of modem data to slot 5 on AC link using out DMA1 |

| Modem Command - RW - 32 bits - [MEM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Modem send out through DMA2 enable | 3 | 0b | Enables sending of modem data to slot 10 on AC link using out DMA2 |
| Modem send out through DMA3 enable | 4 | 0b | Enables sending of modem data to slot 11 on AC link using out DMA3 |
| Status To Mem | 5 | 0b | When reg0x04[1] is enabled, after finishing an input or output modem DT data block, the status will be updated in either DT memory or in reg0x00.<br>0: Update status to register<br>1: Update status to memory |
| Modem_slots_alloc_1 | 6 | 0h | Refer to Modem_slots_alloc_0 (bit 27 of this register) for more detailed information. |
| Reserved | 7 | 0h | |
| In DMA enable | 8 | 0b | Enables input DMA |
| Out DMA #1 enable | 9 | 0b | Enables output DMA 1 |
| Out DMA #2 enable | 10 | 0b | Enables output DMA 2 |
| Out DMA #3 enable | 11 | 0b | Enables output DMA 3 |
| Reserved | 19:12 | 00h | |
| Audio present | 20 | 0b | 1 - Audio is present;<br>0 - Audio is not present (for Modem driver), set by BIOS [Read-only] |
| Reserved | 21 | 0b | |
| Gpio through DMA en | 22 | 0b | 1 - Gpio data (slot 12) comes from DMA 3 along with data for slot 11.<br>0 - Gpio data (slot 12) comes from Gpio register 74h. |
| Loop back enable | 23 | 0b | Enables loop-back mode. The SDATA_OUT connected directly to the SDATA_IN |
| Packed format disable | 24 | 0b | 1 - Disable packed format for sending data: data is sent to memory including data from Slot1 and 2 (physical address and data are sent to memory along with other data).<br>0 - Data sent to memory through DMA in packed format, no slot 1 and 2 - default |
| Burst enable. | 25 | 0b | Enables Burst operation: Will send/request data only based on threshold value & overflow condition. |
| Panic enable | 26 | 0b | Enables panic signal - to get more priority over internal bus. |
| Modem_slots_alloc_0 | 27 | 0b | Once this bit is set to 1, writing 0 to it has no effect.<br><br>{Modem_slots_alloc_1, Modem_slots_alloc_0} =<br>00: Only slot 5 is allocated to the modem.<br>01: Slots 5 and 12 are allocated to the modem.<br>10: Slots 5, 10, 11, and 12 are allocated to the modem.<br>11: Slots 5, 10, 11, and 12 are allocated to the modem. |
| AC Link active | 28 | 0b | 1 - AC Link is in the ACTIVE state.;<br>0 - AC Link is not in the ACTIVE state. [Read-only] |
| AC'97 software reset | 29 | 0b | 0 – De-assert ac97 software reset asynchronous to BIT_CLK<br>1 – Assert ac97software reset asynchronous to BIT_CLK. Registers in BIT_CLK domain will be reset. |
| AC97 Sync | 30 | 0b | 0 – De-assert AC link's SYNC asynchronous to BIT_CLK<br>1 – Assert AC link's SYNC asynchronous to BIT_CLK |
| AC97 Reset# | 31 | 0b | 1 – De-assert AC link's RESET# asynchronous to BIT_CLK<br>0 – Assert AC link's RESET# asynchronous to BIT_CLK |

| Modem Command - RW - 32 bits - [MEM_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Modem Command Register: Controls the operation of Audio Controller. Value of "1" in bit position enables corresponding function. Value of "0" disables it. | | | |

| Phy Status and Address - RW - 32 bits - [MEM_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Codec ID | 1:0 | 00b | 0 – Master AC97<br>1 – 1st slave AC97<br>2 – 2nd slave AC97<br>3 – Reserved |
| Read/Write request | 2 | 0b | 1 – Read request.<br>0 – Write request |
| Reserved | 7:3 | 0b | |
| Phy out address enable | 8 | 0b | 1 - Enable sending out of Physical address in next frame<br>0 - Physical out address have been sent out; open to get new value.<br>When the bit is 0, writing 1 to it makes it 1. Once it is 1, writing to it is ignored. It will automatically return to 0 when physical address is sent out. |
| Phy out address | 15:9 | 0b | Physical out address (wont be able to write new address until old one is send out - until Physical out address enable is '1') (for address write bit [8] is asserted until data send out) |
| Phy out Data | 31:16 | 0000h | Physical register out data (for write). When Physical out address enabled and Write request is set (Physical status [2]=0) then the value in this register will be send out in slot 2 |
| Phy Status/ Phy out address and data Register: Controls the Physical status of the AC'97 Controller and also has the Physical out address and data to be sent with the next frame. | | | |

| Input Phy Address and Data - R - 32 bits - [MEM_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |
| Phy_in_read_flag | 8 | 0b | It is cleared whenever reg0x0Ch[8]=1 and reg0x0C[2]=1.<br>It is set when input physical data (due to a physical read) arrives. |
| in_Phy_addr | 15:9 | 00h | Physical address from AC'97 Codec |
| Input Phy data | 31:16 | 0000h | Input Physical data from AC'97 Codec. |
| Input Phy address and data Register: | | | |

| Input Ch2 GPIO Data - R - 32 bits - [MEM_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SLOTREQ | 9:0 | 000h | The read only bits [0:9] of this field respectively come from slot1[11:2] of AC link's SDATA_IN0/1/2 OR'ed together (even if slot 1 may not be valid as indicated by slot 0[14]). These bits [0:9] respectively decide whether slot 3~12 of SDATA_OUT is allowed or not.<br>0 – Slot is allowed<br>1 – Slot is allowed<br>Even if a bit in this field being 0 indicates a slot is allowed, the controller further looks at reg0x34[9:0] to finally decide whether the slot is enabled. |
| Reserved | 31:10 | 000000h | |
| SLOTREQ Register: | | | |

| Counter - R - 32 bits - [MEM_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Slot Counter | 3:0 | 0h | The current slot number (0-12) which the AC97 controller handling. |
| Reserved | 7:4 | 0h | |
| Bit Clock Counter | 12:8 | 00h | For the tag slot, the value changes from 0 to F. For other slots the value changes from 0 to 20. |
| Reserved | 31:13 | 00000h | |
| Counter Register: | | | |

| Input FIFO Threshold - RW - 32 bits - [MEM_Reg: 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input Threshold | 4:0 | 00h | Threshold value for the Input Channel's FIFOs. (FIFO size 16x20) |
| Reserved | 31:5 | 0000000h | |
| Input FIFO Threshold Register: Input DMA's FIFOs threshold value. | | | |

| Input DMA Link List Pointer - RW - 32 bits - [MEM_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA LL ptr en | 0 | 0b | Input DMA Link List Pointer enable |
| in DMA LL Pointer | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Input DMA Link List Pointer Register: | | | |

| Input DMA DT Start - R - 32 bits - [MEM_Reg: 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the input DMA |
| Input DMA Discrete Table (DT) Start Pointer Register. | | | |

| Input DMA DT Next - R - 32 bits - [MEM_Reg: 28h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT next | 31:0 | 0000_0000h | Pointer to the next DT for the input DMA |
| Input DMA DT Next Pointer Register. | | | |

| Input DMA DT Current - R - 32 bits - [MEM_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the input DMA |
| Input DMA DT Current Pointer Register. | | | |

| Input DMA DT Size & FIFO Info - R - 32 bits - [MEM_Reg: 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| in DMA DT size | 15:0 | 0000h | Data size of DT for input DMA. |
| in FIFO Used | 20:16 | 00h | Number of filled FIFO entries of input DMA. |
| in FIFO Free | 25:21 | 06h | Number of free FIFO entries of input DMA. |
| in DMA State | 28:26 | 0h | Current state of the in DMA |
| Reserved | 31:20 | 000h | |
| Input DMA DT Size & FIFO info Register: Size of data associated with DT for input channels and number of FIFO entries free and used for the input channels. | | | |

| Output DMA 1/2/3 Threshold – RW - 32 bits - [MEM_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA #1 State | 2:0 | 0h | Current state of out DMA # 1 (Modem 1 for slot 5) [Read-only] |
| out DMA #2 State | 5:3 | 0h | Current state of out DMA # 2 (Modem 2 for slot 10) [Read-only] |
| out DMA #3 State | 8:6 | 0h | Current state of out DMA # 3 (Handset for slot 11) [Read-only] |
| Reserved | 15:9 | 00h | |
| Output Threshold 1 | 20:16 | 00h | Threshold value for the Out DMA 1 FIFO. (FIFO size 4x20) |
| Output Threshold 2 | 25:21 | 00h | Threshold value for the Out DMA 2 FIFO. (FIFO size 4x20) |
| Output Threshold 3 | 30:26 | 00h | Threshold value for the Out DMA 3 FIFO. (FIFO size 4x20) |
| Reserved | 31 | 0b | |
| Output DMA1/2/3 Threshold Register: Configure Slot responsibility for Output channels 0 and also threshold values of all out DMA FIFO's. | | | |

| Output DMA1 Link List Pointer – RW - 32 bits - [MEM_Reg: 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA1 LL ptr en. | 0 | 0b | Output DMA1 Link List Pointer enable |
| out DMA1 LL ptr | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Output DMA1 Link List Pointer Register: | | | |

| Output DMA2 Link List Pointer – RW - 32 bits - [MEM_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA2 LL ptr en. | 0 | 0b | Output DMA2 Link List Pointer enable |
| out DMA2 LL ptr | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Output DMA2 Link List Pointer Register: | | | |

| Output DMA3 Link List Pointer – RW - 32 bits - [MEM_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA3 LL ptr en. | 0 | 0b | Output DMA3 Link List Pointer enable |
| out DMA3 LL ptr | 31:1 | 0000_0000h | Pointer to the start of the Link List - to the first DT. |
| Output DMA3 Link List Pointer Register: | | | |

| Output DMA1 DT Start – R - 32 bits - [MEM_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA1 DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the Output DMA1. |
| Output DMA1 Discrete Table (DT) Start Pointer Register: | | | |

| Output DMA1 DT Next– R - 32 bits - [MEM_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA1 DT next | 31:0 | 0000_0000h | Pointer to the next DT for the Output DMA1 |
| Output DMA1 DT Next Pointer Register: | | | |

| Output DMA1 DT Current– R - 32 bits - [MEM_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA1 DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the Output DMA1. |
| Output DMA1 DT Current Pointer Register: | | | |

| Output DMA2 DT Start– R - 32 bits - [MEM_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA2 DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the Output DMA2. |
| Output DMA2 Discrete Table (DT) Start Pointer Register: | | | |

| Output DMA2 DT Next– R - 32 bits - [MEM_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA2 DT next | 31:0 | 0000_0000h | Pointer to the next DT for the Output DMA2 |
| Output DMA2 DT Next Pointer Register: | | | |

| Output DMA2 DT Current– R - 32 bits - [MEM_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA2 DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the Output DMA2. |
| Output DMA2 DT Current Pointer Register: | | | |

| Output DMA3 DT Start– R - 32 bits - [MEM_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA3 DT start | 31:0 | 0000_0000h | Pointer to the start of data associated with current DT for the Output DMA3. |
| Output DMA3 Discrete Table (DT) Start Pointer Register: | | | |

| Output DMA3 DT Next– R - 32 bits - [MEM_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA3 DT next | 31:0 | 0000_0000h | Pointer to the next DT for the Output DMA3 |
| Output DMA3 DT Next Pointer Register: | | | |

| Output DMA3 DT Current– R - 32 bits - [MEM_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA3 DT current | 31:0 | 0000_0000h | Pointer to the currently accessing memory address for the Output DMA3. |
| Output DMA3 DT Current Pointer Register: | | | |

| Output DMA 1/2 DT Size– R - 32 bits - [MEM_Reg: 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA 1 DT size | 15:0 | 0000h | Data size of DT for Output DMA # 1. |
| out DMA 2 DT size | 31:16 | 0000h | Data size of DT for Output DMA # 2. |
| Output DMA 1/2 DT Size Register: | | | |

| Output DMA 3 DT Size– R - 32 bits - [MEM_Reg: 6Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA 1 DT size | 15:0 | 0000h | Data size of DT for Output DMA # 1. |
| Reserved | 31:16 | 0000h | |
| Output DMA 3 DT Size Register: | | | |

| Output DMA 1/2/3 FIFO Info– R - 32 bits - [MEM_Reg: 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA1 Used | 4:0 | 00h | Number of filled FIFO entries of Output DMA1 (FIFO size 6). |
| out DMA2 Used | 9:5 | 00h | Number of filled FIFO entries of Output DAM2 (FIFO size 6). |

| Output DMA 1/2/3 FIFO Info– R - 32 bits - [MEM_Reg: 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| out DMA3 Used | 14:10 | 00h | Number of filled FIFO entries of Output DMA3 (FIFO size 6). |
| Reserved | 15 | 0b | |
| out DMA1 Free | 20:16 | 00h | Number of free FIFO entries of Output DMA1 (FIFO size 6). |
| out DMA2 Free | 25:21 | 00h | Number of free FIFO entries of Output DMA2 (FIFO size 6). |
| out DMA3 Free | 30:26 | 00h | Number of free FIFO entries of Output DMA3 (FIFO size 6). |
| Reserved | 31 | 0b | |
| Output DMA1/2/3 FIFO Used/Free Register: | | | |

| GPIO Out Data– RW - 32 bits - [MEM_Reg: 74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Gpio Out Data en | 0 | 0b | Enables the sending of GPIO data on the next frame's slot 12. Data comes from bit[20:1] of this register. This bit has higher priority than reg0x08[22]. That means if both this bit and reg0x08[22] are set, then GPIO data comes from bit[20:1] of this register, not from DMA3. Writing 1 to this bit enables this data to be sent out. After some time, when data is sent out, this bit automatically goes back to 0. Bit[20:1] can be written only when this bit is 0. |
| Gpio Out Data | 20:1 | 000h | Gpio Out Data |
| Reserved | 31:21 | 000h | |
| Gpio Out Data Register: Data to the modem. | | | |

| Input GPIO Data– R - 32 bits - [MEM_Reg: 78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input Gpio data | 19:0 | 00000h | Input Gpio(modem) data from AC'97 Codecs (ORed) |
| Input Gpio Codec ID | 22:20 | 0h | ID of Codec who send the Gpio data (001 - Master, 010- Slave1, 100 - Slave2) |
| Reserved | 31:23 | 00h | |
| Input Gpio data Register: Data from the modem. | | | |

| Modem Mirror– RW - 32 bits - [MEM_Reg: 7Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Modem mirror | 31:0 | 0000_0000h | Data written by modem for communication with Audio. |
| Modem mirror Register. | | | |

| Audio Mirror– R - 32 bits - [MEM_Reg: 80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Audio mirror | 31:0 | 0000_0000h | Data written by audio for communication with modem. |
| Audio mirror Register. | | | |

| Modem Fifo Flush– W - 32 bits - [MEM_Reg: 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Output DMA1 Fifo Flush | 0 | 0b | Writing to this bit flushes modem output DMA1 fifo, i.e., the indexes and Used/Free counts are reset. Reading this bit returns 0. |
| Output DMA2 Fifo Flush | 1 | 0b | Writing to this bit flushes modem output DMA2 fifo, i.e., the indexes and Used/Free counts are reset. Reading this bit returns 0. |

| Modem Fifo Flush– W - 32 bits - [MEM_Reg: 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Output DMA3 Fifo Flush | 2 | 0b | Writing to this bit flushes modem output DMA3 fifo, i.e., the indexes and Used/Free counts are reset. Reading this bit returns 0. |
| Input DMA Fifo Flush | 3 | 0b | Writing to this bit flushes modem input DMA fifo, i.e., the indexes and Used/Free counts are reset. Reading this bit returns 0. |
| Reserved | 31:4 | 0000000h | |
| Modem Fifo Flush Register. | | | |

| Modem Phy Semaphore Reg- RW - 8 bits - [MEM_Reg: A8h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Modem Phy semaphore | 0 | 0b | PHY is ready for Modem to access.<br>0 = PHY is not ready for Modem to access.<br>1 = PHY is ready for Modem to access.<br><br>Modem driver can only accesses PHY register when the read back value of this bit is '1' after write a '1' to this bit. It is possible the read back value is '0' after write '1' which mean PHY is not ready for Modem to access. Since PHY can be accessed by both audio & modem controllers, the bit is used by modem driver to establish semaphore handshake with audio driver.<br>Driver is also responsible to clear the bit after its PHY read/write access.<br>Note: The driver can check Mem_reg 0x10[8] which indicates a PHY read access completion. For writing to PHY, the driver can clear the semaphore bit after the write command. |
| Reserved | 7:1 | 00h | |

## 2.6 HD Audio Controllers Registers

Note: Some HD Audio functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these HD Audio functions and the associated registers.

### *2.6.1 HD Audio Controller PCI Configuration Space Registers (Device 20 Function 2)*

The PCI Configuration Space Registers define the operation of the SB600's HD Audio Controller. These registers are accessible only when the HD Audio Controller detects a Configuration Read or Write operation, with its IDSEL asserted.

| Register Name | Offset Address |
|---|---|
| Device ID | 00h |
| Vendor ID | 02h |
| PCI Command | 04h |
| PCI Status | 06h |
| Revision ID | 08h |
| Programming Interface | 09h |
| Sub Class Code | 0Ah |
| Base Class Code | 0Bh |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| BIST | 0Fh |
| Lower Base Address Register | 10h |
| Upper Base Address Register | 14h |
| Subsystem Vendor ID | 2Ch |
| Subsystem ID | 2Eh |
| Capabilities Pointer | 34h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Minimum Grant | 3Eh |
| Maximum Latency | 3Fh |
| Misc Control Register | 42h |
| Interrupt Pin Control Register | 44h |
| Power Management Capability ID | 50h |
| Power Management Capabilities | 52h |
| Power Management Control/Status | 54h |
| MSI Capability ID | 60h |
| MSI Message Control | 62h |
| MSI Message Lower Address | 64h |
| MSI Message Upper Address | 68h |
| MSI Message Data | 6Ch |

| Vendor ID – R – 16 bits – [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | Identifies the vendor as AMD. |

| Device ID – RW – 16 bits – [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Device ID | 15:0 | 4383h | Identifies this device as the HD Audio Controller. |

| PCI Command – RW – 16 bits – [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | Reserved. |
| Memory Space Enable | 1 | 0b | Enables the HD Audio controller to respond to PCI memory space access. |
| Bus Master Enable | 2 | 0b | Enables the HD Audio controller's bus mastering capability. |
| Reserved | 9:3 | 00h | Reserved |
| Interrupt Disable | 10 | 0b | Disables the device from asserting INTx#.<br>Note: This bit does not affect the generation of MSI. |
| Reserved | 15:11 | 00h | Reserved |

| PCI Status – RW – 16 bits – [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 2:0 | 0h | Reserved |
| Interrupt Status | 3 | 0b | This bit is a "1" when INTx# is asserted.<br>Note: This bit is not set by MSI. |
| Capabilities List | 4 | 1b | This bit is hardwired to "1" to indicate that the HD Audio controller contains a capability pointer list. The first item at offset 34h |
| Reserved | 12:5 | 00h | Reserved |
| Received Master Abort | 13 | 0 | When set, this bit indicates that the HD Audio controller terminated a PCI bus operation with a Master Abort. |
| Reserved | 15:14 | 0h | Reserved |

| Revision ID – R – 8 bits – [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | These bits are hardwired to "0" to indicate the revision level of the chip design (for the SB600). |

| Programming Interface – R – 8 bits – [PCI_Reg: 09h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Programming Interface | 7:0 | 00h | Programming Interface. |

| Sub Class Code – R – 8 bits – [PCI_Reg: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Sub Class Code | 7:0 | 03h | Sub Class Code. Indicates a HD Audio device in the context of a multimedia device class. |

| Base Class Code – R – 8 bits – [PCI_Reg: 0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Class Code | 7:0 | 04h | Base Class Code. Indicates a multimedia device. |

| Cache Line Size – RW – 8bits – [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | This field is implemented as a read/write field for legacy compatibility purposes only and has no functional impact. |

| Latency Timer – R – 8 bits – [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer | 7:0 | 00h | Hardwired to “0”. |

| Header Type – R – 8 bits – [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 00h | Hardwired to “0”. |

| BIST – R – 8 bits – [PCI_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST | 15:0 | 0000h | Hardwired to “0”. |

| Lower Base Address Register – RW – 32 bits – [PCI_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Space Type | 0 | 0b | Hardwired to “0” to indicate this BAR is located in memory space only. |
| Address Range | 2:1 | 10b | Hardwired to 10b to indicate this BAR can be located anywhere in 64-bit address space. |
| Prefetchable | 3 | 0b | Hardwired to “0” to indicate this BAR is not prefetchable. |
| Reserved | 13:4 | 000h | Hardwired to “0”. |
| Lower Base Address | 31:14 | 00000h | Lower Base Address for the HD Audio controller’s memory mapped configuration registers. 16K bytes are requested by hardwiring bits[13:4] to 0. |

| Upper Base Address Register – RW – 32 bits – [PCI_Reg: 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Upper Base Address | 31:0 | 00000000h | Upper Base Address for the HD Audio controller’s memory mapped configuration registers. |

| Subsystem Vendor ID – RW – 16 bits – [PCI_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | This register is implemented as write-once register. Any subsequent writes have no effect. |

| Subsystem ID – RW – 16 bits – [PCI_Reg: 2Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem ID | 15:0 | 0000h | This register is implemented as write-once register. Any subsequent writes have no effect. |

| Capabilities Pointer – R – 8 bits – [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 50h | This register indicates the offset for the capability pointer |

| Interrupt Line – RW – 8 bits – [PCI_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | This register is used to communicate to software the interrupt line that the interrupt pin is connected to. It is not used by the HD Audio controller. |

| Interrupt Pin – R – 8 bits – [PCI_Reg: 3Dh] | | | |
|---|---|---|---|
| Interrupt Pin | 3:0 | 1h | This register reflects the value programs into Interrupt Control Pin register at offset 44h, bits[3:0] |
| Reserved | 7:4 | 0h | Reserved |

| Minimum Grant – R – 8 bits – [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minimum Grant | 7:0 | 00h | Hardwired to “0”. |

| Maximum Latency – R – 8 bits – [PCI_Reg: 3Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Maximum Latency | 7:0 | 00h | Hardwired to “0”. |

| Misc Control Register – RW – 8 bits – [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Disable Non-noop | 0 | 0b | 1: Non-snoop attribute is disabled on Buffer Descriptor and Data Buffer DMA.<br>0: Set the Non-snoop attribute on Buffer Descriptor and Data Buffer DMA when the Traffic Priority bit is set in the Stream Descriptor. |
| Disable Non-snoop Override | 1 | 0b | 1: Bit[0] of this register controls the Non-snoop attribute<br>0: override the bit[0] setting meaning always generate No Snoop attribute on Buffer Descriptor and Data Buffer DMA |
| Enable Non-snoop Request | 2 | 0b | 1: Enable Non-snoop request to ACPI<br>0: Disable Non-snoop request to ACPI<br>When enabled and the DMA cycle is Non-snoop, ACPI will not generate a wake to CPU in C2 state. |
| Reserved | 7:3 | 00h | Reserved |

| Interrupt Pin Control Register – RW – 8 bits – [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin Control | 3:0 | 1h | Controls the value reports in Interrupt Pin Register at offset 0x3D. |
| Reserved | 7:4 | 0h | Reserved |

| Power Management Capability ID – R – 16 bits – [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 01h | Hardwired to 01h. Indicates PCI Power Management Capability. |
| Next Capability Pointer | 15:8 | 60h | Hardwired to 60h.<br>Next capability is at offset 60h |

| Power Management Capabilities – R – 16 bits – [PCI_Reg: 52h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Version | 2:0 | 010b | Hardwired to 010b. Indicates this function complies with Revision 1.1 of the PCI Power Management Interface Specification |
| PME Clock | 3 | 0b | Hardwired to “0”. Indicates that no PCI clock is required for the function to generate PME#. |
| Reserved | 4 | 0b | Reserved |
| Device Specific Initialization | 5 | 0b | Hardwired to “0”. Indicates that no device specific initialization is required. |
| Aux Current | 8:6 | 001b | Hardwired to 001b. Indicates 55mA maximum suspend well current is required in the D3cold state. |
| D1 Support | 9 | 0b | Hardwired to “0”. Indicates D1 state is not supported. |
| D2 Support | 10 | 0b | Hardwired to “0”. Indicates D2 state is not supported. |
| PME Support | 15:11 | 11001b | Hardwired to 11001b. Indicates PME# can be generated from D0 and D3 states. |

| Power Management Control/Status – RW – 32 bits – [PCI_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Power State | 1:0 | 0h | This field is used both to determine the current power state and to set a new power state of the HD Audio controller. |
| Reserved | 7:2 | 00h | Reserved |
| PME Enable | 8 | 0b | Enables the function to assert PME#.<br>This bit is in resume well and only cleared on power-on reset. |
| Reserved | 14:9 | 00h | Reserved |
| PME Status | 15 | 0b | This bit set when HD Audio controller asserts the PME# signal, it is independent of the PME Enable bit.<br>Writing a “1” clears this bit.<br>This bit is in resume well and only cleared on power-on reset. |
| Reserved | 31:16 | 0000h | Reserved |

| MSI Capability ID – R – 16 bits – [PCI_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capability ID | 7:0 | 05h | Hardwired to 05h. Indicates MSI Capability. |
| Next Capability Pointer | 15:8 | 00h | Hardwired to “0”. Indicates this is the last capability structure in the list. |

| MSI Message Control – RW - 16 bits – [PCI_Reg: 62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Enable | 0 | 0b | Enables MSI if set to “1”. |
| Multiple Message Capable | 3:1 | 0h | Hardwired to “0”. Indicates support for one message only. |
| Multiple Message Enable | 6:4 | 0h | Hardwired to “0”. Indicates support for one message only. |
| 64 Bit Address Capability | 7 | 1b | Hardwired to “1”. Indicates the ability to generate 64-bit message address. |
| Reserved | 15:8 | 00h | Reserved |

| MSI Message Lower Address – RW - 32 bits – [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Message Lower Address | 31:2 | 00000000h | Lower Address used for MSI Message. |
| Reserved | 01:0 | 0h | Reserved |

| MSI Message Upper Address – RW - 32 bits – [PCI_Reg: 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Message Upper Address | 31:0 | 00000000h | Upper Address used for MSI Message. |

| MSI Message Data – RW - 16 bits – [PCI_Reg: 6Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Message Data | 15:0 | 0000h | Data used for MSI Message. |

### *2.6.2 HD Audio Controller Memory Mapped Registers*

The base memory location for these memory mapped registers is specified in the PCI Configuration Upper and Lower Base Address Registers. The individual registers are then accessible at Base + offset as indicated in the following table. These registers are accessed in byte, word, or dword quantities.

| Register Name | Address Offset |
|---|---|
| Global Capabilities | 00h |
| Minor Version | 02h |
| Major Version | 03h |
| Output Payload Capability | 04h |
| Input Payload Capability | 06h |
| Global Control | 08h |
| Wake Enable | 0Ch |
| State Change Status | 0Eh |
| Global Status | 10h |
| Output Stream Payload Capability | 18h |
| Input Stream Payload Capability | 1Ah |
| Interrupt Control | 20h |
| Interrupt Status | 24h |
| Wall Clock Counter | 30h |
| Stream Synchronization | 38h |
| CORB Lower Base Address | 40h |
| CORB Upper Base Address | 44h |
| CORB Write Pointer | 48h |
| CORB Read Pointer | 4Ah |
| CORB Control | 4Ch |
| CORB Status | 4Dh |
| CORB SIze | 4Eh |
| RIRB Lower Base Address | 50h |
| RIRB Upper Address | 54h |
| RIRB Write Pointer | 58h |
| RIRB Response Interrupt Control | 5Ah |
| RIRB Control | 5Ch |
| RIRB Status | 5Dh |
| RIRB Size | 5Eh |
| Immediate Command Output Interface | 60h |

| Register Name | Address Offset |
|---|---|
| Immediate Command Input Interface | 64h |
| Immediate Command Input Interface | 68h |
| DMA Position Buffer Lower Base Address | 70h |
| DMA Position Buffer Upper Base Address | 74h |
| Control | 80h |
| Status | 83h |
| Link Position in Current Buffer | 84h |
| Cyclic Buffer Length | 88h |
| Last Valid Index | 8Ch |
| FIFO Size | 90h |
| Stream Format | 92h |
| Buffer Descriptor Lower Base Address | 98h |
| Buffer Descriptor Upper Base Address | 9Ch |
| Control | A0h |
| Status | A3h |
| Link Position in Current Buffer | A4h |
| Cyclic Buffer Length | A8h |
| Last Valid Index | ACh |
| FIFO Size | B0h |
| Stream Format | B2h |
| Buffer Descriptor Lower Base Address | B8h |
| Buffer Descriptor Upper Base Address | BCh |
| Control | C0h |
| Status | C3h |
| Link Position in Current Buffer | C4h |
| Cyclic Buffer Length | C8h |
| Last Valid Index | CCh |
| FIFO Size | D0h |
| Stream Format | D2h |
| Buffer Descriptor Lower Base Address | D8h |
| Buffer Descriptor Upper Base Address | DCh |
| Control | E0h |
| Status | E3h |
| Link Position in Current Buffer | E4h |
| Cyclic Buffer Length | E8h |
| Last Valid Index | ECh |
| FIFO Size | F0h |
| Stream Format | F2h |
| Buffer Descriptor Lower Base Address | F8h |
| Buffer Descriptor Upper Base Address | FCh |
| Control | 100h |
| Status | 103h |
| Link Position in Current Buffer | 104h |
| Cyclic Buffer Length | 108h |
| Last Valid Index | 10Ch |
| FIFO Size | 110h |
| Stream Format | 112h |
| Buffer Descriptor Lower Base Address | 118h |
| Buffer Descriptor Upper Base Address | 11Ch |
| Control | 120h |
| Status | 123h |
| Link Position in Current Buffer | 124h |
| Cyclic Buffer Length | 128h |
| Last Valid Index | 12Ch |
| FIFO Size | 130h |
| Stream Format | 132h |
| Buffer Descriptor Lower Base Address | 138h |
| Buffer Descriptor Upper Base Address | 13Ch |

| Register Name | Address Offset |
|---|---|
| Control | 140h |
| Status | 143h |
| Link Position in Current Buffer | 144h |
| Cyclic Buffer Length | 148h |
| Last Valid Index | 14Ch |
| FIFO Size | 150h |
| Stream Format | 152h |
| Buffer Descriptor Lower Base Address | 158h |
| Buffer Descriptor Upper Base Address | 15Ch |
| Control | 160h |
| Status | 163h |
| Link Position in Current Buffer | 164h |
| Cyclic Buffer Length | 168h |
| Last Valid Index | 16Ch |
| FIFO Size | 170h |
| Stream Format | 172h |
| Buffer Descriptor Lower Base Address | 178h |
| Buffer Descriptor Upper Base Address | 17Ch |
| Wall Clock Counter Alias | 2030h |
| Input Stream Descriptor 0 - Link Position in Current Buffer Alias | 2084h |
| Input Stream Descriptor 1 - Link Position in Current Buffer Alias | 20A4h |
| Input Stream Descriptor 2 - Link Position in Current Buffer Alias | 20C4h |
| Input Stream Descriptor 3 - Link Position in Current Buffer Alias | 20E4h |
| Output Stream Descriptor 0 - Link Position in Current Buffer Alias | 2104h |
| Output Stream Descriptor 1 - Link Position in Current Buffer Alias | 2124h |
| Output Stream Descriptor 2 - Link Position in Current Buffer Alias | 2144h |
| Output Stream Descriptor 3 - Link Position in Current Buffer Alias | 2164h |

| Global Capabilities – R – 16 bits - [Mem_Reg: Base + 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| 64 Bit Address Supported | 0 | 1b | Hardwired to “1”. Indicates that 64-bit addressing capability is supported by the HD Audio controller for BDL, Data Buffer, Command Buffer, and Response Buffer addresses. |
| Number of Serial Data Output Signals | 2:1 | 0h | Hardwired to “0”. Indicates that one SDO line is supported. |
| Number of Bidirectional Streams Supported | 7:3 | 0h | Hardwired to “0”. Indicates that bidirectional stream is not supported. |
| Number of Input Streams Supported | 11:8 | 4h | 4 Input Streams are supported. |
| Number of Output Streams Supported | 15:12 | 4h | 4 Output Streams are supported. |

| Minor Version – R – 8 bits - [Mem_Reg: Base + 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minor Version | 7:0 | 00h | Hardwired to “0”. |

| Major Version – R – 8 bits - [Mem_Reg: Base + 03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Major Version | 7:0 | 01h | Hardwired to 01h. |

| Output Payload Capability – R – 16 bits - [Mem_Reg: Base + 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Output Payload Capability | 15:0 | 003Ch | Hardwired to 3Ch. Indicates the total output payload on the link. This does not include bandwidth used for command and control. This measurement is in 16-bit word quantities per 48kHz frame. The default link clock speed of 24MHz (double data rate) provides 1000 bits per frame minus 40 bits for command and control, leaving 960 bits (60 words) for data payload. |

| Input Payload Capability – R – 16 bits - [Mem_Reg: Base + 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input Payload Capability | 15:0 | 001Dh | Hardwired to 1Dh. Indicates the total input payload on the link. This does not include bandwidth used for response. This measurement is in 16-bit word quantities per 48kHz frame. The default link clock speed of 24MHz provides 500 bits per frame minus 36 bits for response, leaving 464 bits (29 words) for data payload. |

| Global Control – RW – 32 bits - [Mem_Reg: Base + 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Controller Reset | 0 | 0b | Writing a 0 to this bit causes the controller to transition to the Reset state. After the hardware has completed sequencing into the Reset state, it will report a 0 in this bit. Software must read a 0 from this bit to verify that the controller is in reset.<br>Writing a “1” to this bit causes the controller to exit the Reset state and deassert the link RESET# signals. Software is responsible for setting/clearing this bit such that the minimum link RESET# signal assertion pulse width specification is met.<br>When the controller hardware is ready to begin operation, it will report a “1” in this bit. Software must read a 1 from this bit before accessing any controller registers. |
| Flush Control | 1 | 0b | Writing a 1 to this bit initiates a flush. The flush is completed when Flush Status is set. |
| Reserved | 7:2 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| Accepted Unsolicited Response Enable | 8 | 0b | If “1”, Unsolicited Response from the codecs are accepted by the controller and placed into the Response Input Ring Buffer. If “0”, Unsolicited Responses are accepted and dropped. |
| Reserved | 31:9 | 000000h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| Wake Enable – RW – 16 bits - [Mem_Reg: Base + 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Wake Enable | 3:0 | 0h | This field controls which SDIN signals may generate a wake event in response to a codec State Change event.<br>Bit[0] corresponds to Codec 0 – SDIN[0]<br>Bit[1] corresponds to Codec 1 – SDIN[1]<br>Bit[2] corresponds to Codec 2 – SDIN[2]<br>Bit[3] corresponds to Codec 3 – SDIN[3]<br>These bits are only cleared by a power-on reset. |
| Reserved | 15:4 | 000h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| State Change Status – RW – 16 bits - [Mem_Reg: Base + 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| State Change Status Flags | 3:0 | 0h | This field indicates which SDIN signal(s) received a State Change event.<br>Bit[0] corresponds to Codec 0 – SDIN[0]<br>Bit[1] corresponds to Codec 1 – SDIN[1]<br>Bit[2] corresponds to Codec 2 – SDIN[2]<br>Bit[3] corresponds to Codec 3 – SDIN[3]<br>These bits are cleared by writing 1's to them.<br>These bits are only cleared by a power-on reset. |
| Reserved (R/W) | 15:4 | 000h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| Global Status – RW – 16 bits - [Mem_Reg: Base + 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | Reserved. Software must use 0 for write to this bit. |
| Flush Status | 1 | 0b | This bit is set to a "1" by hardware to indicate that the flush cycle initiated by setting the Flush Control has completed. Software must write a "1" to clear this bit before the next time Flush Control is set. |
| Reserved | 15:2 | 0000h | Reserved. Software must use 0's for write to these bits. |

| Output Stream Payload Capability – R – 16 bits - [Mem_Reg: Base + 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Output Stream Payload Capability | 15:0 | 003Ch | This field indicates the maximum number of words per frame for any single output stream. This measurement is in 16-bit word quantities per 48 kHz frame. The value must not be greater than the Output Payload Capability register value. Software must ensure that a format which would cause more words per frame than indicated is not programmed into the Output Stream Descriptor register.<br>00h: No Limit ( Stream size is limited only by Output Payload Capability register)<br>01h: 1 word payload<br>:<br>:<br>FFh: 255 word payload |

| Input Stream Payload Capability – R – 16 bits - [Mem_Reg: Base + 1Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input Stream Payload Capability | 15:0 | 0000h | This field indicates the maximum number of words per frame for any single input stream. This measurement is in 16-bit word quantities per 48 kHz frame. The value must not be greater than the Input Payload Capability register value. Software must ensure that a format which would cause more words per frame than indicated is not programmed into the Input Stream Descriptor register.<br>00h: No Limit ( Stream size is limited only by Input Payload Capability register)<br>01h: 1 word payload<br>:<br>:<br>FFh: 255 word payload |

| Interrupt Control – RW – 32 bits - [Mem_Reg: Base + 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Stream Interrupt Enable | 7:0 | 00h | When set to “1”, the individual streams are enabled to generate an interrupt when the corresponding stream status bits are set.<br>A stream interrupt is caused as a result of a buffer with IOC in the BDL entry being completed or as result of FIFO error. Control over the generation of each of these sources is in the associated Stream Descriptor.<br>Bit[0]: Input Stream 0<br>Bit[1]: Input Stream 1<br>Bit[2]: Input Stream 2<br>Bit[3]: Input Stream 3<br>Bit[4]: Output Stream 0<br>Bit[5]: Output Stream 1<br>Bit[6]: Output Stream 2<br>Bit[7]: Output Stream 3 |
| Reserved | 29:8 | 000000h | Reserved |
| Controller Interrupt Enable | 30 | 0b | Enables the general interrupt for controller functions. When set to “1’, the controller generates an interrupt when the corresponding status bit gets set due to a Response Interrupt, a Response Buffer Overrun, and Wake events. |
| Global Interrupt Enable | 31 | 0b | Enables device interrupt generation. When set to “1”, the HD Audio device is enabled to generate an interrupt. This control is in addition to any bits in the bus specific address space such as the Interrupt Enable bit in the PCI Configuration Space. |

| Interrupt Status – RW – 32 bits - [Mem_Reg: Base + 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Stream Interrupt Status | 7:0 | 00h | A "1" indicates that an interrupt condition occurred on the corresponding stream. These bits are cleared by writing 1's to them.<br>Note that these bits are set regardless of the state of the corresponding Interrupt Enable bits. |
| Reserved | 29:8 | 000000h | Reserved |
| Controller Interrupt Status | 30 | 0b | A "1" indicates that an interrupt condition occurred. This bit is cleared by writing a "1" to it.<br>Note that this bit is set regardless of the state of the corresponding Interrupt Enable bit. |
| Global Interrupt Status | 31 | 0b | This bit is an "OR" of all the interrupt status bits in this register. |

| Wall Clock Counter – R – 32 bits - [Mem_Reg: Base + 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Wall Clock Counter | 31:0 | 00000000h | 32 bit counter that Is incremented at the link bitclock rate and rolls over from FFFF_FFFFh to 0000_0000h. This counter will roll over to 0 with period of approximately 179 seconds with the nominal 24 MHz bitclock rate. |

| Stream Synchronization – RW – 32 bits – [Mem_Reg: Base + 38h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Stream Synchronization | 7:0 | 00h | When these bits are set, they block data from being sent on or received from the link. Each bit controls the associated Stream Descriptor.<br>To synchronously start a set of DMA engines, the bits in this register are set to a "1". The RUN bits for the associated Stream Descriptors can be set to a "1" to start the DMA engines. When all streams are ready, the associated Stream Synchronization bits can all be set to 0 at the same time, and transmission or reception from the link will begin together at the start of the next full link frame.<br>To synchronously stop streams, these bits are set, and the RUN bits in the Stream Descriptors are cleared by software. |
| Reserved | 31:8 | 000000h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| CORB Lower Base Address – RW – 32 bits – [Mem_Reg: Base + 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Lower Base Address Unimplemented Bits | 6:0 | 00h | Hardwired to 0. This forces 128-byte buffer alignment for cache line fetch optimizations. |
| CORB Lower Base Address | 31:7 | 0000000h | Upper 25 bits of the 32 bits Lower Base address of the Command Output Ring Buffer, allowing the CORB Base Address to be assigned on any 1 KB boundary. This register must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

| CORB Upper Base Address – RW – 32 bits – [Mem_Reg: Base + 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Upper Base Address | 31:0 | 00000000h | Upper 32 bits address of the CORB. This register must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

| CORB Write Pointer – RW – 16 bits – [Mem_Reg: Base + 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Write Pointer | 7:0 | 00h | Software writes the last valid CORB entry offset into this field in dword granularity. The DMA engine fetches commands from the CORB until the Read Pointer matches the Write Pointer. This supports up to 256 CORB entries. This field may not be written while the DMA engine is running. |
| Reserved | 15:8 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| CORB Read Pointer – RW – 16 bits – [Mem_Reg: Base + 4Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Read Pointer | 7:0 | 00h | Software reads this field to determine how many commands it can write to the CORB without over-running. The value read indicates the CORB Read Pointer offset in dword granularity. The offset entry read from this field has been successfully fetched by the DMA controller and may be over-written by software. This field may be read while the DMA engine is running. |
| Reserved | 14:8 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| CORB Read Pointer Reset | 15 | 0b | Software writes a 1 to this bit to reset the CORB Read Pointer to 0 and clear any residual prefetched commands in the CORB hardware buffer within the controller. The hardware will physically update this bit to 1 when the CORB pointer reset is complete. Software must read a 1 to verify that the reset completed correctly. Software must clear this bit back to 0, by writing a 0, and then read back the 0 to verify that the clear completed correctly. The CORB DMA engine must be stopped prior to resetting the Read Pointer or else DMA transfer may be corrupted. |

| CORB Control – RW – 8 bits – [Mem_Reg: Base + 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Memory Error Interrupt Enable | 0 | 0b | If this bit is set, the controller will generate an interrupt if the Memory Error Interrupt bit is set. |
| Enable CORB DMA Engine | 1 | 0b | 0: DMA Stop<br>1: DMA Run<br>After software writes a “0” to this bit, the hardware may not stop immediately. The hardware will physically update the bit to “0” when the DMA engine is truly stopped. Software must read a “0” from this bit to verify that the DMA engine is truly stopped. |
| Reserved | 7:2 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| CORB Status – RW – 8 bits – [Mem_Reg: Base + 4Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Memory error Indication | 0 | 0b | If this status bit is set, the controller has detected an error in the pathway between the controller and memory. Writing a "1" to this bit will clear the bit, but a CRST must be performed before operation continues/ |
| Reserved | 7:2 | 00h | Reserved. Software must use 0's for write to these bits. |

| CORB Size – RW – 8 bits – [Mem_Reg: Base + 4Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CORB Size | 1:0 | 10b | These bits have no functional impact to the hardware. This HD Audio controller only supports 256 entries. |
| Reserved | 3:2 | 0h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| CORB Size Capability | 7:4 | 0100b | Hardwired to 0100b indicating this controller only supports a CORB size of 256 entries. |

| RIRB Lower Base Address – RW – 32 bits – [Mem_Reg: Base + 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RIRB Lower Base Address Unimplemented Bits | 6:0 | 00h | Hardwired to 0. This forces 128-byte buffer alignment for cache line fetch optimizations. |
| RIRB Lower Base Address | 31:7 | 0000000h | Upper 25 bits of the 32 bits Lower Base Address of the Response Input Ring Buffer, allowing the RIRB Base Address to be assigned on any 2 KB boundary. This register must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

| RIRB Upper Address – RW – 32 bits – [Mem_Reg: Base + 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RIRB Upper Base Address | 31:0 | 00000000 h | Upper 32 bits address of the RIRB. This register must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

| RIRB Write Pointer – RW – 16 bits – [Mem_Reg: Base + 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RIRB Write Pointer | 7:0 | 00h | This field indicates the last valid RIRB entry written by the DMA controller. Software reads this field to determine how many responses it can read from the RIRB. The value read indicates the RIRB Write Pointer offset in two dwords since each RIRB entry is two dwords. This field may be read while the DMA engine is running. |
| Reserved | 14:8 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| RIRB Write Pointer Reset | 15 | 0b | Software writes a "1" to this bit to reset the RIRB Write Pointer to 0's. The DMA engine must be stopped prior to resetting the Write Pointer or else DMA transfer may be corrupted. This bit will always be read as 0. |

| RIRB Response Interrupt Count – RW – 16 bits – [Mem_Reg: Base + 5Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| N Response Interrupt Count | 7:0 | 00h | 01h = 1 Response sent to RIRB<br>:<br>FFh = 255 Responses sent to RIRB<br>00h = 256 Responses sent to RIRB<br>The DMA engine should be stopped when changing this field or else an interrupt may be lost.<br>Note that each Response occupies two dwords in the RIRB.<br>If more than one codec responds in one frame, then the count is increased by the number of responses received in the frame. |
| Reserved | 15:8 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| RIRB Control – RW – 8 bits – [Mem_Reg: Base + 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Response Interrupt Control | 0 | 0b | 0 = Disable Interrupt<br>1 = Generate an interrupt after N number of Responses are sent to the RIRB buffer or when an empty Response slot is encountered on all SDIN_x inputs after a frame which return a response (whichever occurs first). The N counter is reset when the interrupt is generated. |
| RIRB DMA Enable | 1 | 0b | 0 = DMA Stop<br>1 = DMA Run |
| Response Overrun Interrupt Control | 2 | 0b | If this bit is set, the hardware will generate an interrupt when the Response Overrun Interrupt Status is set. |
| Reserved | 7:3 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| RIRB Status – RW – 8 bits – [Mem_Reg: Base + 5Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Response Interrupt | 0 | 0b | Hardware sets this bit to “1” when an interrupt has been generated after N number of responses are sent to the RIRB buffer, or when empty Response slot is encountered on all SDIN_x inputs (whichever occurs first). Software clears this bit by writing a “1” to this bit. |
| Reserved | 1 | 0b | Reserved. Software must use 0’s for write to these bits. |
| Response Overrun Interrupt Status | 2 | 0b | Hardware sets this bit to a “1’ when an overrun occurs in the RIRB. An interrupt may be generated if the Response Overrun Interrupt Control bit is set.<br>Software clears this bit by writing a “1” to it. |
| Reserved | 7:3 | 0h | Reserved. Software must use 0’s for write to these bits. |

| RIRB Size – RW – 8 bits – [Mem_Reg: Base + 5Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RIRB Size | 1:0 | 10b | These bits have no functional impact to the hardware. This HD Audio controller only supports 256 entries. |
| Reserved | 3:2 | 0h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| RIRB Size Capability | 7:4 | 4h | Hardwired to 4h indicating this controller only supports a RIRB size of 256 entries. |

| Immediate Command Output Interface – RW – 32 bits – [Mem_Reg: Base + 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Immediate Command Write | 31:0 | 00000000 h | The value written into this register is used as the verb to be sent out over the link when the ICB (Immediate Command Busy) bit is set to "1". Software must ensure that the ICB bit is cleared before writing a value into this register or undefined behavior will result. Reads from this register will always return 0's. |

| Immediate Command Input Interface – RW – 32 bits – [Mem_Reg: Base + 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Immediate Response Read | 31:0 | 00000000 h | This register contains the value from the last response to come in over the link.<br>If multiple codecs respond in the same frame, which one of the responses that will be saved is indeterminate. |

| Immediate Command Input Interface – RW – 16 bits – [Mem_Reg: Base + 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Immediate Command Status | 0 | 0b | This bit is a "0" when the controller can accept an immediate command. Software must wait for this bit to be 0 before writing a value in the ICW register.<br>This bit will be clear (indicating "ready") when the following conditions are met: (1) the link is running, (2) the CORB is not active (CORBRP = CORBWP or CORBEN is not set), and (3) there is not an immediate command already in the queue waiting to be sent.<br>Writing this bit to "1" will cause the contents of the ICW register to be sent as a verb in the next frame. Once a response is received the IRV bit will be set and this bit will be cleared indicating ready to transmit another verb. |
| Immediate Result Valid | 1 | 0b | This bit is set to a "1" by hardware when a new response is latched into the IRR (Immediate Response Read) register. Software must clear this bit before issuing a new command by writing a one to it so that the software may determine when a new response has arrived. |
| Reserved | 2 | 0b | Reserved. Software must use 0's for write to these bits. |
| Immediate Response Result Unsolicited | 3 | 0b | Indicates whether the response latched in the Immediate Response Input Register is a solicited or unsolicited response. |
| Immediate Response Result Address | 7:4 | 00h | The address of the codec which sent the response currently latched into the Immediate Response Input. |
| Reserved | 15:8 | 0000h | Reserved. Software must use 0's for write to these bits. |

| DMA Position Lower Base Address – RW – 32 bits – [Mem_Reg: Base + 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA Position Buffer Enable | 0 | 0b | When this bit is set to a "1', the controller will write the DMA positions of each of the DMA engines to the buffer in main memory periodically. Software can use this value to know what data in memory is valid data. |
| DMA Position Lower Base Address Unimplemented Bits | 6:1 | 00h | Hardwired to 0. This forces 128-byte buffer alignment for cache line fetch optimizations. |

| DMA Position Lower Base Address – RW – 32 bits – [Mem_Reg: Base + 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA Position Lower Base Address | 31:7 | 0000000h | Contains the upper 25 bits of the lower 32 bits of the DMA Position Buffer Base Address.<br>This same address is used by the Flush Control, and must be programmed with a valid value before the Flush Control is initiated. |

| DMA Position Upper Base Address – RW – 32 bits – [Mem_Reg: Base + 74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA Position Upper Base Address | 31:0 | 00000000h | Upper 32 bits of the DMA Position Buffer Base Address.<br>This same address is used by the Flush Control, and must be programmed with a valid value before the Flush Control is initiated. |

| Stream Descriptor Control – RW – 24 bits<br>Input Stream 0 - [Mem_Reg: Base + 80h]<br>Input Stream 1 - [Mem_Reg: Base + A0h]<br>Input Stream 2 - [Mem_Reg: Base + C0h]<br>Input Stream 3 - [Mem_Reg: Base + E0h]<br>Output Stream 0 - [Mem_Reg: Base + 100h]<br>Output Stream 1 - [Mem_Reg: Base + 120h]<br>Output Stream 2 - [Mem_Reg: Base + 140h]<br>Output Stream 3 - [Mem_Reg: Base + 160h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Stream Reset | 0 | 0b | Writing a “1” causes the corresponding stream to be reset. The Stream Descriptor registers (except this bit), FIFO’s and cadence generator for the corresponding stream are reset. After the stream hardware has completed sequencing into the reset state, it will report a “1” in this bit. Software must read a “1’ from this bit to verify that the stream is in reset.<br>Writing a “0” causes the corresponding stream to exit reset. When the stream hardware is ready to begin operation, it will report a “0” in this bit. Software must read a “0” from this bit before accessing any of the stream registers. The Run bit must be cleared before asserting SRST (Stream Reset). |
| Stream Run | 1 | 0b | When set to “1”, the DMA engine associated with this input stream will be enabled to transfer data in the FIFO to main memory. When cleared to “0”, the DMA engine associated with this input stream will be disabled. If the corresponding SSYNC bit is “0”, input stream data will be taken from the link and moved to the FIFO and an over-run may occur. |
| Interrupt On Completion Enable | 2 | 0b | Controls whether an interrupt is generated when the Buffer Completion Interrupt Status is set |
| FIFO Error Interrupt Enable | 3 | 0b | Controls whether an interrupt is generated when the FIFO Error is set. |
| Descriptor Error Interrupt Enable | 4 | 0b | Controls whether an interrupt is generated when the Descriptor Error Status is set. |
| Reserved | 15:5 | 000h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| Stream Descriptor Control – RW – 24 bits<br>Input Stream 0 - [Mem_Reg: Base + 80h]<br>Input Stream 1 - [Mem_Reg: Base + A0h]<br>Input Stream 2 - [Mem_Reg: Base + C0h]<br>Input Stream 3 - [Mem_Reg: Base + E0h]<br>Output Stream 0 - [Mem_Reg: Base + 100h]<br>Output Stream 1 - [Mem_Reg: Base + 120h]<br>Output Stream 2 - [Mem_Reg: Base + 140h]<br>Output Stream 3 - [Mem_Reg: Base + 160h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Stripe Control | 17:16 | 0h | The hardware only supports one SDO, this field has no functional impact. |
| Traffic Priority | 18 | 0b | If set to “1”, it will cause the controller to generate non-snooped traffic. |
| Bidirectional Direction Control | 19 | 0b | The hardware does not support bi-direction, this field has no impact. |
| Stream Number | 23:20 | 0h | The value reflects the Tag associated with the data being transferred on the link.<br>When data controlled by this descriptor is sent out over the link, it will have this stream number encoded on the SYNC signal.<br>When an input stream is detected on any of the SDIN_x signals that match this value, the data are loaded into the FIFO associated with this descriptor.<br>0000b = Reserved<br>0001b = Stream 1<br>:<br>1111b = Stream 15 |

| Stream Descriptor Status – RW – 8 bits<br>Input Stream 0 - [Mem_Reg: Base + 83h]<br>Input Stream 1 - [Mem_Reg: Base + A3h]<br>Input Stream 2 - [Mem_Reg: Base + C3h]<br>Input Stream 3 - [Mem_Reg: Base + E3h]<br>Output Stream 0 - [Mem_Reg: Base + 103h]<br>Output Stream 1 - [Mem_Reg: Base + 123h]<br>Output Stream 2 - [Mem_Reg: Base + 143h]<br>Output Stream 3 - [Mem_Reg: Base + 163h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 0h | Reserved. Software must use 0's for write to these bits. |
| Buffer Completion Interrupt Status | 2 | 0b | For an Output Stream engine, this bit is set to "1" by the hardware after the last byte of data has been fetched from memory and put into DMA FIFO and the current descriptor has the IOC bit set.<br>For an Input Stream engine, this bit is set to "1" by the hardware after the last byte of data has been removed from the DMA FIFO and the current descriptor has the IOC bit set.<br>This bit is cleared by writing a "1" to this bit. |
| FIFO Error | 3 | 0b | Set when a FIFO error occurs regardless of the FIFO Error Interrupt Enable bit.<br>This bit is cleared by writing a "1" to this bit. |
| Descriptor Error | 4 | 0b | During the fetch of a descriptor, an error has occurred. |
| FIFO Ready | 5 | 0b | For an Output Stream, the controller hardware will set this bit to a "1" while the output DMA FIFO contains enough data to maintain the stream on the link. |
| Reserved | 7:6 | 0h | Reserved. Software must use 0's for write to these bits. |

| Stream Descriptor Link Position in Buffer – R – 32 bits<br>Input Stream 0 - [Mem_Reg: Base + 84h]<br>Input Stream 1 - [Mem_Reg: Base + A4h]<br>Input Stream 2 - [Mem_Reg: Base + C4h]<br>Input Stream 3 - [Mem_Reg: Base + E4h]<br>Output Stream 0 - [Mem_Reg: Base + 104h]<br>Output Stream 1 - [Mem_Reg: Base + 124h]<br>Output Stream 2 - [Mem_Reg: Base + 144h]<br>Output Stream 3 - [Mem_Reg: Base + 164h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Link Position in Buffer | 31:0 | 00000000h | This field indicates the number of bytes that have been received off the link. |

| Stream Descriptor Cyclic Buffer Length – RW – 32 bits<br>Input Stream 0 - [Mem_Reg: Base + 88h]<br>Input Stream 1 - [Mem_Reg: Base + A8h]<br>Input Stream 2 - [Mem_Reg: Base + C8h]<br>Input Stream 3 - [Mem_Reg: Base + E8h]<br>Output Stream 0 - [Mem_Reg: Base + 108h]<br>Output Stream 1 - [Mem_Reg: Base + 128h]<br>Output Stream 2 - [Mem_Reg: Base + 148h]<br>Output Stream 3 - [Mem_Reg: Base + 168h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cyclic Buffer Length | 31:0 | 00000000h | Indicates the number of bytes in the complete cyclic buffer. Link Position in Buffer will be reset when it reaches this value.<br>Software may only write to this register after Global Reset, Controller Reset, or Stream Reset has occurred. Once the Run bit has been set to enable the engine, software must not write to this register until after the next reset is asserted, or undefined events will occur. |

| Stream Descriptor Last Valid Index – RW – 16 bits<br>Input Stream 0 - [Mem_Reg: Base + 8Ch]<br>Input Stream 1 - [Mem_Reg: Base + ACh]<br>Input Stream 2 - [Mem_Reg: Base + CCh]<br>Input Stream 3 - [Mem_Reg: Base + ECh]<br>Output Stream 0 - [Mem_Reg: Base + 10Ch]<br>Output Stream 1 - [Mem_Reg: Base + 12Ch]<br>Output Stream 2 - [Mem_Reg: Base + 14Ch]<br>Output Stream 3 - [Mem_Reg: Base + 16Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Last Valid Index | 7:0 | 00h | The value written to this register indicates the index for the last valid Buffer Descriptor in BDL. After the controller has processed this descriptor, it will wrap back to the first descriptor in the list and continue processing.<br>LVI (Last Valid Index) must be “1”; i.e., there must be at least two valid entries in the BDL before DMA operations can begin.<br>This value should not be modified except when the Run bit is “0”. |
| Reserved | 15:8 | 00h | Reserved. Software must do a read-modify-write to preserve the value of these bits. |

| Stream Descriptor FIFO Size – R – 16 bits<br>Input Stream 0 - [Mem_Reg: Base + 90h]<br>Input Stream 1 - [Mem_Reg: Base + B0h]<br>Input Stream 2 - [Mem_Reg: Base + D0h]<br>Input Stream 3 - [Mem_Reg: Base + F0h]<br>Output Stream 0 - [Mem_Reg: Base + 110h]<br>Output Stream 1 - [Mem_Reg: Base + 130h]<br>Output Stream 2 - [Mem_Reg: Base + 150h]<br>Output Stream 3 - [Mem_Reg: Base + 170h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FIFO Size | 15:0 | 0000h | For Output Stream, the FIFO Size varies between 32 dwords to 256 dwords depending on the Stream Format. For Input Stream, the FIFO Size is fixed at 64 dwords. |

| Stream Descriptor Format – RW – 16 bits<br>Input Stream 0 - [Mem_Reg: Base + 92h]<br>Input Stream 1 - [Mem_Reg: Base + B2h]<br>Input Stream 2 - [Mem_Reg: Base + D2h]<br>Input Stream 3 - [Mem_Reg: Base + F2h]<br>Output Stream 0 - [Mem_Reg: Base + 112h]<br>Output Stream 1 - [Mem_Reg: Base + 132h]<br>Output Stream 2 - [Mem_Reg: Base + 152h]<br>Output Stream 3 - [Mem_Reg: Base + 172h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Number of Channels | 3:0 | 0h | Number of channels in each frame of the stream.<br>0000b = 1<br>0001b = 2<br>:<br>1111b = 16 |
| Bits per Sample | 6:4 | 0h | 000b = 8 bits<br>001b = 16 bits<br>010b = 20 bits<br>011b = 24 bits<br>100b = 32 bits<br>101b – 111b = Reserved |
| Reserved | 7 | 0b | Reserved. Software must do a read-modify-write to preserve the value of these bits. |
| Sample Base Rate Divisor | 10:8 | 0h | 000b = Divide by 1 (48 kHz, 44.1 kHz)<br>001b = Divide by 2 (24 kHz, 22.05 kHz)<br>010b = Divide by 3 (16 kHz, 32 kHz)<br>011b = Divide by 4 (11.025 kHz)<br>10b0 = Divide by 5 (9.6 kHz)<br>101b = Divide by 6 (8 kHz)<br>110b = Divide by 7<br>111b = Divide by 8 (6 kHz) |
| Sample Base Rate Multiple | 13:11 | 0h | 000b = 1 (48 kHz, 44.1 kHz)<br>001b = 2 (96 kHz, 88.2 kHz, 32 kHz)<br>010b = 3 (144 kHz)<br>011b = 4 (192 kHz, 176.4 kHz)<br>101b – 111b = Reserved |
| Sample Base Rate | 14 | 0b | 0 = 48 kHz<br>1 = 44.1 kHz |
| Reserved | 15 | 0b | Reserved |

| Stream Descriptor BDL Pointer Lower Base Address – RW – 32 bits<br>Input Stream 0 - [Mem_Reg: Base + 98h]<br>Input Stream 1 - [Mem_Reg: Base + B8h]<br>Input Stream 2 - [Mem_Reg: Base + D8h]<br>Input Stream 3 - [Mem_Reg: Base + F8h]<br>Output Stream 0 - [Mem_Reg: Base + 138h]<br>Output Stream 2 - [Mem_Reg: Base + 158h]<br>Output Stream 3 - [Mem_Reg: Base + 178h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 6:0 | 00h | Hardwired to 0's to force 128 byte alignment of the BDL. |
| Buffer Descriptor List Lower Base Address | 31:7 | 0000000h | Upper 25 bits of the lower 32 bit address of the Buffer Descriptor List. This value should not be modified except when the Run bit is "0". |

| Stream Descriptor BDL Pointer Upper Base Address – RW – 32 bits<br>Input Stream 0 - [Mem_Reg: Base + 9Ch]<br>Input Stream 1 - [Mem_Reg: Base + BCh]<br>Input Stream 2 - [Mem_Reg: Base + DCh]<br>Input Stream 3 - [Mem_Reg: Base + FCh]<br>Output Stream 0 - [Mem_Reg: Base + 11Ch]<br>Output Stream 1 - [Mem_Reg: Base + 13Ch]<br>Output Stream 2 - [Mem_Reg: Base + 15Ch]<br>Output Stream 3 - [Mem_Reg: Base + 17Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Buffer Descriptor List Upper Base Address | 31:0 | 00000000h | Upper 32 bit address of the Buffer Descriptor List. This value should not be modified except when the Run bit is "0". |

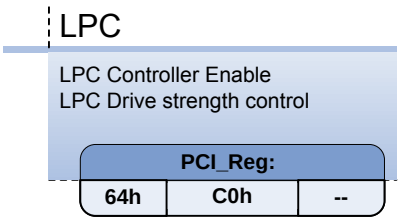
| Wall Clock Counter Alias – R – 32 bits – [Mem_Reg: Base + 2030h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Wall Clock Counter Alias | 31:0 | 00000000h | An alias of the Wall Clock Counter register at offset 30h. It behaves exactly the same as if the Wall Clock Counter register were being read directly. |

| Stream Descriptor Link Position in Buffer Alias – R – 32 bits<br>Input Stream 0 - [Mem_Reg: Base + 2084h]<br>Input Stream 1 - [Mem_Reg: Base + 20A4h]<br>Input Stream 2 - [Mem_Reg: Base + 20C4h]<br>Input Stream 3 - [Mem_Reg: Base + 20E4h]<br>Output Stream 0 - [Mem_Reg: Base + 2104h]<br>Output Stream 1 - [Mem_Reg: Base + 2124h]<br>Output Stream 2 - [Mem_Reg: Base + 2144h]<br>Output Stream 3 - [Mem_Reg: Base + 2164h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Link Position in Buffer Alias | 31:0 | 00000000h | An alias of the Link Position in Buffer register of each Stream Descriptor. |

# 3 Register Descriptions: PCI Bridges

## 3.1 LPC ISA Bridge (Device 20, Function 3)

Note: Some LPC functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these LPC functions and the associated registers.




### *3.1.1 Programming Interface*

- Write LPC_enable bit in function 0, register 64, bit 20.
- Enable LPC address decode ranges.
- Program DMA controller for any bus master or DMA cycles.
- Perform LPC cycles from PCI or DMA requests from LPC agent.

### *3.1.2 PCI Configuration Registers*

The LPC host controller supports a set of configuration register required by the PCI specification.

Registers not listed here will have this default behavior: read from the register returns 0's; write to the register is ignored.

| Register Name | Offset Address |
|---|---|
| VID | 00h |
| DID | 02h |
| CMD | 04h |
| STATUS | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| BIST | 0Fh |
| Base Address Reg 0 | 10h |
| Subsystem ID & Subsystem Vendor ID | 2Ch |
| Capabilities Pointer | 34h |
| PCI Control | 40h |
| IO Port Decode Enable Register 1 | 44h |
| IO Port Decode Enable Register 2 | 45h |
| IO Port Decode Enable Register 3 | 46h |
| IO Port Decode Enable Register 4 | 47h |
| IO/Mem Port Decode Enable Register 5 | 48h |
| LPC Sync Timeout Count | 49h |
| IO/Mem Port Decode Enable Register 6 | 4Ah |
| Memory Range Register | 4Ch |
| Rom Protect 0 | 50h |

| Register Name | Offset Address |
|---|---|
| Rom Protect 1 | 54h |
| Rom Protect 2 | 58h |
| Rom Protect 3 | 5Ch |
| PCI Memory Start Address for LPC Target Cycles | 60h |
| PCI Memory End Address for LPC Target Cycles | 62h |
| PCI IO base Address for Wide Generic Port | 64h |
| LPC ROM Address Range 1 (Start Address) | 68h |
| LPC ROM Address Range 1 (End Address) | 6Ah |
| LPC ROM Address Range 2 (Start Address) | 6Ch |
| LPC ROM Address Range 2 (End Address) | 6Eh |
| Firmware Hub Select | 70h |
| Alternative Wide Io Range Enable | 74h |
| Reserved | 78h |
| TPM register | 7Ch |
| MSI Capability register | 80h |

PCI function 3 configuration registers are described below.

| VID- R - 16 bits - [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vendor ID | 15:0 | 1002h | Vendor ID |
| Vendor ID Register: This register holds a unique 16-bit value assigned to a vendor, and combined with the device ID it identifies any PCI device. | | | |

| DID- R - 16 bits - [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Devide ID | 15:0 | 438Dh | Device ID |
| Device ID Register: This register holds a unique 16-bit value assigned to a device and together with the vendor ID, it identifies any PCI device. | | | |

| CMD- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Space | 0 | 1b | Hardcoded to 1 to enable IO access, since legacy IOs reside behind the LPC bridge. |
| Memory Space | 1 | 1b | Hardcoded to 1 to enable Memory Access, since BIOS ROM is located behind the LPC bridge. |
| Bus Master | 2 | 1b | Hardcoded to 1 to enable bus master, since LPC bridge handles legacy DMA cycles. |
| Special Cycles | 3 | 1b | Hardcoded to 1 to enable Special Cycle recognition, since special cycle must be recognized by the LPC bridge. |
| Memory Write and Invalidate Enable | 4 | 0b | Hardcoded to 0 to indicate that Memory Write and Invalidate command is not implemented. |
| VGA Palette Snoop | 5 | 0b | Hardcoded to 0 to indicate that VGA Palette Snoop is disabled - The LPC bridge does not need to snoop VGA palette cycles. |
| Parity Error Response | 6 | 0b | PERR# (Response) Detection Enable bit.<br>1 - LPC bridge asserts PERR# when it is the agent receiving data AND it detects a parity error.<br>0 – LPC bridge does not assert PERR#. |
| Stepping Control | 7 | 0b | Hardcoded to 0 to indicate that LPC bridge does not use perform address/data stepping - the LPC bridge does not need to insert a wait state between the address and data on the AD lines. |

| CMD- RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SERR# Enable | 8 | 0b | SERR# enable - If set to 1, the LPC bridge asserts SERR# when it detects as address parity error. SERR# is not asserted if this bit is 0. |
| Fast Back-tp-Back Enable | 9 | 0b | Hardcoded to 0 to indicate that Fast Back-to-back is disabled. The LPC bridge only acts as a master to a single device, so this functionality is not needed. |
| Reserved | 15:10 | 00h | |
| Command Register: The PCI specification defines this register to control a PCI device's ability to generate and respond to PCI cycles. | | | |

| STATUS- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | |
| Capabilities List | 4 | 0b | This bit is read only. When reg0x78[1] is 1, this bit reads 1; when reg0x78[1] is 0, this bit reads 0. This is to satisfy K8 requirement. For P4 system, 0x78[1] should be set to 0 to mask off this bit. |
| Reserved | 7:5 | 0h | |
| Master Data Parity Error | 8 | 0h | Data Parity reported – Set to 1 if the LPC bridge detects PERR# asserted while acting as PCI master (whether PERR# was driven by the LPC bridge or not.) |
| Device Select Timing | 10:9 | 1h | DEVSEL# timing – Read only bits indicating DEVSEL# timing when performing a positive decode.<br>Since DEVSEL# is asserted to meet the medium timing, these bits are encoded as 01b. |
| Signaled Target Abort | 11 | 0b | Signaled Target Abort – This bit is set to 1 when the LPC bridge signals Target Abort. |
| Received Target Abort | 12 | 0b | Received Target Abort – This bit is set to 1 when the LPC bridge -generated PCI cycle is aborted by a PCI target. Cleared by writing a 1 to it. |
| Received Master Abort | 13 | 0b | Received Master Abort Status. Set to 1 when the LPC bridge acts as a PCI master and aborts a PCI bus cycle. Cleared by writing a 1 to this bit. |
| Signaled System Error | 14 | 0b | SERR# status. This bit is set to 1, when the LPC bridge detects a system error. |
| Detected Parity Error | 15 | 0b | Detected Parity Error. This bit is set to 1 when the LPC bridge detects a parity error. |
| Status Register: The PCI specification defines this register to record status information for PCI related events. Writing 0 to a bit has no effect; writing 1 to a bit will clear it to 0. | | | |

| Revision ID/Class Code - R - 8 bits - [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | These bits are hardwired to 00h to indicate the revision level of the chip design |
| Class Code | 31:8 | 060100h | Class Code. |
| Revision ID/Class Code Register: This read only register contains the device's revision information and generic function. Since SB600 is an ISA bridge, its assigned class code is 060100h. | | | |

| Cache Line Size - R - 8 bits - [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | Cache Line Size. |
| Cache Line Size Register: This register specifies the system cache line size. This register is not implemented. | | | |

| Latency Timer - R - 8 bits - [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Latency Timer | 7:0 | 00h | Latency Timer. |
| Latency Timer Register: This register specifies the value of the Latency Timer in units of PCICLKs. | | | |

| Header Type - R - 8 bits - [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 80h | Header Type. |
| Header Type Register: This register identifies the type of the predefined header in the configuration space. Since THE SB600 is a multifunction device, the most significant bit is set. | | | |

| BIST- R - 8 bits - [PCI_Reg: 0Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| BIST | 7:0 | 00h | BIST. |
| Built-in Self Test Register: This register is used for control and status for Built-in Self Test. LPC has no BIST modes. | | | |

| Base Address Reg 0- RW* - 32 bits - [PCI_Reg: 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Base Address 0 | 31:0 | FEC0_0000h | Base address register 0. |
| This register is write-only. Reading it always returns 0000_0000h. It has an internal value used as base address for APIC memory space. Writing to the register will change its internal value, but only bit[31:5] is overwritten, and bit[4:0] is hardwired to 00000b. The default internal value is FEC0_0000h. | | | |

| Subsystem ID & Subsystem Vendor ID – RW - 32 bits - [PCI_Reg: 2Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Subsystem Vendor ID | 15:0 | 0000h | Subsystem Vendor ID. |
| Subsystem ID | 31:16 | 0000h | Subsystem ID. |
| This 4-byte register is a write-once & read-only afterward register. The BIOS writes this register once (all 4 bytes at once) and software reads its value (when needed). | | | |

| Capabilities Pointer - R - 32 bits - [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 00h | When reg0x78[1] (msi on) is 0, this field reads 0; when reg0x78[1] is 1, this field reads 80h, pointing to the starting address of MSI capability register |
| Reserved | 31:8 | 000000h | |

| PCI Control - RW - 8 bits - [PCI_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 0h | |
| DMA Enable | 2 | 0b | Setting it to 1 enables lpc DMA cycle. Note: 32-bit DMA is not supported. Transfer size: Channels 0-3: 8 bits, channels 5-7: 16 bits. |
| Reserved | 7:3 | 00h | |

| IO Port Decode Enable Register 1- RW - 8 bits - [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Parallel Port Enable 0 | 0 | 0b | Port enable for parallel port, 378-37fh |
| Parallel Port Enable 1 | 1 | 0b | Port enable for parallel port, 778-77fh |
| Parallel Port Enable 2 | 2 | 0b | Port enable for parallel port, 278-27fh |
| Parallel Port Enable 3 | 3 | 0b | Port enable for parallel port, 678-67fh |
| Parallel Port Enable 4 | 4 | 0b | Port enable for parallel port, 3bc-3bfh |

| IO Port Decode Enable Register 1- RW - 8 bits - [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Parallel Port Enable 5 | 5 | 0b | Port enable for parallel port, 7bc-7bfh |
| Serial Port Enable 0 | 6 | 0b | Port enable for serial port, 3f8-3ffh |
| Serial Port Enable 1 | 7 | 0b | Port enable for serial port, 2f8-2ffh |
| This register controls the decoding of parallel & serial ports. Writing '1' to a bit enables the corresponding IO range. | | | |

| IO Port Decode Enable Register 2- RW - 8 bits - [PCI_Reg: 45h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Serial Port Enable 2 | 0 | 0b | Port enable for serial port, 220-227h |
| Serial Port Enable 3 | 1 | 0b | Port enable for serial port, 228-22fh |
| Serial Port Enable 4 | 2 | 0b | Port enable for serial port, 238-23fh |
| Serial Port Enable 5 | 3 | 0b | Port enable for serial port, 2e8-2efh |
| Serial Port Enable 6 | 4 | 0b | Port enable for serial port, 338-33fh |
| Serial Port Enable 7 | 5 | 0b | Port enable for serial port, 3e8-3efh |
| Audio Port Enable 0 | 6 | 0b | Port enable for audio port, 230-233h (Range 220-22fh needs to be enabled using bits 0 and 1) |
| Audio Port Enable 1 | 7 | 0b | Port enable for audio port, 240-253h |
| This register controls the decoding of serial & audio ports. Writing '1' to a bit enables the corresponding IO range. | | | |

| IO Port Decode Enable Register 3- RW - 8 bits - [PCI_Reg: 46h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Audio Port Enable 2 | 0 | 0b | Port enable for audio port, 260-273h |
| Audio Port Enable 3 | 1 | 0b | Port enable for audio port, 280-293h |
| MIDI Port Enable 0 | 2 | 0b | Port enable for MIDI port, 300-301h |
| MIDI Port Enable 1 | 3 | 0b | Port enable for MIDI port, 310-311h |
| MIDI Port Enable 2 | 4 | 0b | Port enable for MIDI port, 320-321h |
| MIDI Port Enable 3 | 5 | 0b | Port enable for MIDI port, 330-331h |
| MSS Port Enable 0 | 6 | 0b | Port enable for MSS port, 530-537h |
| MSS Port Enable 1 | 7 | 0b | Port enable for MSS port, 604-60bh |
| This register controls the decoding of audio, MIDI, & MSS ports. Writing '1' to a bit enables the corresponding IO range. | | | |

| IO Port Decode Enable Register 4- RW - 8 bits - [PCI_Reg: 47h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSS Port Enable 2 | 0 | 0b | Port enable for MSS port, e80-e87h |
| MSS Port Enable 3 | 1 | 0b | Port enable for MSS port, f40-f47h |
| FDC Port Enable 0 | 2 | 0b | Port enable for FDC port, 3f0-3f7h |
| FDC Port Enable 1 | 3 | 0b | Port enable for FDC port, 370-377h |
| Game Port Enable | 4 | 0b | Port enable for Game port, 200-20fh |
| KBC Port Enable | 5 | 0b | Port enable for KBC port, 60 & 64h |
| ACPI Micro-Controller Port Enable | 6 | 0b | Port enable for ACPI Micro-Controller port, 62 & 66h |
| Ad-Lib Port Enable | 7 | 0b | Port enable for Ad-Lib port, 388-389h |
| This register controls the decoding of MSS, FDC, game, KBC, ACPI micro-controller, & Ad-lib ports. Writing '1' to a bit enables the corresponding IO range. | | | |

| IO/Mem Port Decode Enable Register 5- RW - 8 bits - [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Super IO Configuration Port Enable | 0 | 0b | Port enable for Super IO config port, 2e-2fh |
| Alternate Super IO Configuration Port Enable | 1 | 0b | Port enable for Alternate super IO config port, 4e-4fh |
| Wide Generic IO Port Enable | 2 | 0b | Port enable for wide generic port, see register 64-65h |

| IO/Mem Port Decode Enable Register 5- RW - 8 bits - [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rom Range 1 Port Enable | 3 | 0b | Port enable for LPC ROM address range 1 (memory), see register 68-6bh |
| Rom Range 2 Port Enable | 4 | 0b | Port enable for LPC ROM address range 2 (memory), see register 6c-6fh |
| Memory Range Port Enable | 5 | 0b | Port enable for LPC memory target range, see register 60-63h |
| RTC IO Range Port Enable | 6 | 0b | Port enable for RTC I/O range 70h~73h |
| Sync Timeout Counter Enable | 7 | 1b | LPC sync timeout counter enabled when set to '1' (default), otherwise the counter is disabled. This counter is used to avoid a deadlock condition if an LPC device drives sync forever. Timeout count is programmed in register 49h. Write '0' to this bit if an LPC device is extremely slow & takes more than 255 LPC clocks to complete a cycle. |
| This register controls the decoding of Super IO configuration, alternate super-IO configuration, wide generic ports, ROM1 & ROM2 ports, and memory port. Writing '1' to a bit enables the corresponding IO/ROM/Memory range. Bit 7 controls enable/disable of LPC sync timeout counter. | | | |

| LPC Sync Timeout Count - RW - 8 bits - [PCI_Reg: 49h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Sync Timeout Count | 7:0 | FFh | Sync Timeout Count. |
| This register contains the value of LPC sync timeout count. This is the number of LPC clocks the state machine will wait when LPC data = sync before aborting the cycle (when timeout counter is enabled; see register 48, bit 7). | | | |

| IO/Mem Port Decode Enable Register 6- RW - 8 bits - [PCI_Reg: 4Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO port enable 0 | 0 | 0b | Port enable for IO port 400h-43Fh |
| IO port enable 1 | 1 | 0b | Port enable for IO port 480h-4BFh |
| IO port enable 2 | 2 | 0b | Port enable for IO port 500h-53Fh |
| IO port enable 3 | 3 | 0b | Port enable for IO port 580h-5BFh |
| Mem port enable | 4 | 0b | Port enable for 4K byte memory range defined in reg0x4C |
| IO port enable 4 | 5 | 0b | Port enable for IO port 80h |
| IO port enable 5 | 6 | 0b | Port enable for IO port 4700h-470Bh |
| IO port enable 6 | 7 | 0b | Port enable for IO port FD60h-FD6Fh |

| IO/Mem Port Decode Enable Register 7- RW - 8 bits - [PCI_Reg: 4Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Wide_io1_enable | 0 | 0b | Wide IO port 1 (defined in register 66/67h) enable |
| Wide_io2_enable | 1 | 0b | Wide IO port 2 (defined in register 90/91h) enable |
| Reserved | 7:2 | 00h | |

| Memory Range Register - RW - 32 bits - [PCI_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 11:0 | 000h | |
| Base Address | 31:12 | 00000h | This register defines a 4K byte memory range from {Base Address, 000h} to {Base Address, ffffh}. The range is enabled by reg0x4A[4]. |

| Rom Protect 0 - RW - 32 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Write Protect | 0 | 0b | When this bit is set, the memory range defined by this register is write-protected. Writing to the range has no effect. |
| Read Protect | 1 | 0b | When this bit is set, the memory range defined by this register is read-protected. Reading any location in the range returns FFFF_FFFFh. |
| Rom Offset | 10:2 | 000h | Rom range offset |
| Rom Base | 31:11 | 000000h | Rom Base and Rom Offset together define the ROM range to be protected. The range is:<br>From {Rom Base, 000_0000_0000b} to {Rom Base, 000_0000_0000b} + {0_0000_0000_0000b, Rom Offset, 11_1111_1111b} |
| This register is write/read, but it can only be written once after hardware reset. Subsequent writes to it have no effect. | | | |

| Rom Protect 1 - RW - 32 bits - [PCI_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| This register has exactly the same definition as that of Rom Protect 0. | | | |

| Rom Protect 2 - RW - 32 bits - [PCI_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| This register has exactly the same definition as that of Rom Protect 0. | | | |

| Rom Protect 3 - RW - 32 bits - [PCI_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| This register has exactly the same definition as that of Rom Protect 0. | | | |

| PCI Memory Start Address for LPC Target Cycles - RW - 16 bits - [PCI_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Memory Start Address | 15:0 | 0000h | 16-bit starting address of the LPC target (memory) range. |
| This register contains upper 16-bits of the starting address of the LPC memory target range. The lower 16-bits of the starting address are considered 0's. This range can be enabled/disabled using register 48h, bit 5. | | | |

| PCI Memory End Address for LPC Target Cycles - RW - 16 bits - [PCI_Reg: 62h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Memory End Address | 15:0 | 0000h | 16-bit END address of the LPC target (memory) range. |
| This register contains upper 16-bits of the ending address of the LPC memory target range. The lower 16-bits of the end address are considered 1's. This range can be enabled/disabled using register 48h, bit 5. | | | |

| PCI IO base Address for Wide Generic Port - RW - 16 bits - [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Base Address 0 | 15:0 | 0000h | 16-bit PCI IO base address for wide generic port range. 512 byte wide range. This function is enabled by PCI register 48h, bit 2 |
| IO Base Address 1 | 31:16 | 0000h | 16-bit PCI IO base address for wide generic port range. 512 byte wide range. This function is enabled by PCI register 4Bh, bit 0 |
| This register contains two 16-bits of IO base address for LPC IO (wide generic port) target range. The limit address is found by adding 512 to the base address. | | | |

| LPC ROM Address Range 1 (Start Address) - RW - 16 bits - [PCI_Reg: 68h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rom Start Address 1 | 15:0 | 0eh (if iLpc_Rom strap is enabled),<br>00h (if the strap is disabled) | 16-bit starting address of the LPC ROM (memory) address range 1. |
| This register contains upper 16-bits of the starting address of the LPC ROM address range 1. The lower 16-bits of the starting address are considered 0's. This range can be enabled by setting register 48h, bit 3 or when LPC ROM is chosen by strap pin. | | | |

| LPC ROM Address Range 1 (End Address) - RW - 16 bits - [PCI_Reg: 6Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rom End Address 1 | 15:0 | 0fh (if iLpc_Rom strap is enabled),<br>00h (if the strap is disabled) | 16-bit END address of the LPC ROM (memory) address range 1. |
| This register contains upper 16-bits of the ending address of the LPC ROM address range 1. The lower 16-bits of the end address are considered 1's. This range can be enabled by setting register 48h, bit 3 or when LPC ROM is chosen by strap pin. | | | |

| LPC ROM Address Range 2 (Start Address)- RW - 16 bits - [PCI_Reg: 6Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rom Start Address 2 | 15:0 | FFFEh (if iLpc_Rom strap is enabled),<br>00h (if the strap is disabled) | 16-bit starting address of the LPC ROM (memory) address range 2. |
| This register contains upper 16-bits of the starting address of the LPC ROM address range 2. The lower 16-bits of the starting address are considered 0's. This range can be enabled by setting register 48h, bit 4 or when LPC ROM is chosen by strap pin. | | | |

| LPC ROM Address Range 2 (End Address) - RW - 16 bits - [PCI_Reg: 6Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rom End Address 2 | 15:0 | FFFFh (if iLpc_Rom strap is enabled),<br>00h (if the strap is disabled) | 16-bit END address of the LPC ROM (memory) address range 2. |
| This register contains upper 16-bits of the ending address of the LPC ROM address range 2. The lower 16-bits of the end address are considered 1's. This range can be enabled by setting register 48h, bit 4 or when LPC ROM is chosen by strap pin. | | | |

| Firmware Hub Select – RW* - 32 bits - [PCI_Reg: 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FWH_C0_IDSEL | 3:0 | 7h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFC0 0000h-FFC7 FFFFh<br>FF80 0000h-FF87 FFFFh |
| FWH_C8_IDSEL | 7:4 | 6h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFC8 0000h-FFCF FFFFh<br>FF88 0000h-FF8F FFFFh |
| FWH_D0_IDSEL | 11:8 | 5h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFD0 0000h-FFD7 FFFFh<br>FF90 0000h-FF97 FFFFh |

| Firmware Hub Select – RW* - 32 bits - [PCI_Reg: 70h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| FWH_D8_IDSEL | 15:12 | 4h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFD8 0000h-FFDF FFFFh<br>FF98 0000h-FF9F FFFFh |
| FWH_E0_IDSEL | 19:16 | 3h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFE0 0000h-FFE7 FFFFh<br>FFA0 0000h-FFA7 FFFFh |
| FWH_E8_IDSEL | 23:20 | 2h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFE8 0000h-FFEF FFFFh<br>FFA8 0000h-FFAF FFFFh |
| FWH_F0_IDSEL | 27:24 | 1h | IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFF0 0000h-FFF7 FFFFh<br>FFB0 0000h-FFB7 FFFFh |
| FWH_F8_IDSEL | 31:28 | 0h | Read only.<br>IDSEL for two 512 KB FWH memory ranges and one 128KB memory range. This field is fixed at 0000. The IDSEL in this field addresses the following memory ranges:<br>FFF8 0000h-FFFF FFFFh<br>FFB8 0000h-FFBF FFFFh<br>000E 0000h-000F FFFFh |
| This register is used to generate the 4-bit IDSEL phase when LPC does read/write to firmware hub memory on LPC bus. If PCI address falls in certain range, the 4-bit value in the register for that range is used as IDSEL. Every firmware hub memory on LPC bus has 4 strap pins. If value on those pins matches IDSEL from LPC host, the memory is selected.<br>*bit [31:28] read only<br>Default Value: 01234567h | | | |

| Alternative Wide IO Range Enable- RW - 32 bits - [PCI_Reg: 74h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Alternative Wide Io Range Enable | 0 | 0b | Wide IO range is usually 512 bytes. With this bit set, the range changes to 16 bytes only. To use this feature, address in reg0x64~65 must be aligned to 16 bytes, i.e., bit[3:0] must be 0. If the address is not aligned to 16bytes, the IO range is from address [15:0] to {address [15:4], 0xF}. |
| Reserved | 1 | 0h | |
| Alternative Wide Io 1 Range Enable | 2 | 0h | Similar to bit 0, but it applies to the IO address defined in reg0x66~67. |
| Alternative Wide Io 2 Range Enable | 3 | 0h | Similar to bit 0, but it applies to the IO address defined in reg0x90~91. |
| Reserved | 7:4 | 00h | |

| Miscellaneous Control Bits- RW - 8 bits - [PCI_Reg: 78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DMA_Enhance | 0 | 1b | 1—Turn on the enhancement feature for the DMA function. This is for better bus efficiency<br>0—Enhancement off. |

| Miscellaneous Control Bits- RW - 8 bits - [PCI_Reg: 78h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Msi On | 1 | 0b | When this bit is set to 1, it turns on LPC MSI capability. The following will be true:<br>* Reg0x04[20] (capabilities list) reads 1.<br>* Reg0x34[7:0] (capabilities pointer) reads 80h<br>When this bit is set to 0, it turns off lpc MSI capability. The following will be true:<br>* Reg0x04[20] reads 0.<br>* Reg0x34[7:0] reads 0. |
| Reserved | 31:2 | 0000_0000h | |

| TPM (trusted plant form module) register- RW - 32 bits - [PCI_Reg: 7Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Tpm12_en | 0 | 0b | When set to 1, it enables decoding of tpm (trusted platform module) cycles defined in TPM1.2 spec (Refer to the addresses defined in bit[1] below). Note that tpm12_en and tpm_legacy are independent bits; they respectively turn on decoding of different tpm addresses. |
| Tpm_amd | 1 | 0b | This bit is replaced with strap pin K8system (to support AMD K8 CPU), and no longer in use. It is read only and returns 0.<br><br>When the strap is 0, it ONLY supports these normal tpm cycles. Here are the cycle definition. (Left hand side is system/software memory address, which is translated to LPC IO address on the right hand side.)<br>0xFED4_0xxx --> 0x0xxx<br>0xFED4_1xxx --> 0x1xxx<br>0xFED4_2xxx --> 0x2xxx<br>0xFED4_3xxx --> 0x3xxx<br>0xFED4_4xxx --> 0x4xxx<br>When the strap is 1, it ONLY supports these AMD tpm cycles.<br>0xFED4_0xxx --> 0x0xxx<br>0xFED4_1xxx --> 0x1xxx<br>0xFED4_2xxx --> 0x2xxx<br>0xFED4_3xxx --> 0x3xxx<br>0xFD_F920_0000~0xFD_F923_FFFF --> 0x4028<br>0xFD_F928_0000~0xFD_F928_0003 --> 0x4020<br>0xFD_F928_0004~0xFD_F928_0007 --> 0x4024~0x4027 |
| Tpm_legacy | 2 | 0b | When set to 1, it enables decoding of legacy tpm addresses, i.e., IO addresses 7E/7F and EE/EF will be decoded. |
| Tmkbc_enable | 3 | 0b | Enable bit for the TMKBC function |
| Tmkbc_set | 4 | 0b | Write once bit. Once set, all tmkbc address/remap registers cannot be changed until the next reset. |
| Tmkbc_sel | 6:5 | 0h | There are actually four sets of TMKBC mapping registers. These two bits select which one of four sets of tmkbc registers at 84, 88, and 8Ch to be accessed. |
| Reserved | 31:7 | 0000000h | |
| Note: Any tpm cycle above is decoded only when the cycle is started by ALinkBridge. Access from bus master devices is not allowed. | | | |

| MSI Capability Register- R - 32 bits - [PCI_Reg: 80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CAP ID | 7:0 | 08h | CAP ID. |
| CAP Next Pointer | 15:8 | 00h | CAP Next Pointer. |
| CAP Enable | 16 | 1b | CAP Enable. |

| MSI Capability Register- R - 32 bits - [PCI_Reg: 80h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| CAP Fixed | 17 | 1b | CAP Fixed. |
| Reserved | 26:18 | 000h | |
| CAP Type | 31:27 | 15h | CAP Type. |

| TMKBC_BaseAddrLow Register- RW - 32 bits - [PCI_Reg: 84h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 1:0 | 00b | |
| Addr64 | 2 | 0b | Defines whether the address is 32 or 64 bits. When this bit = 1, the address is 64 bits; 0 means 32 bits. |
| MaskBits10thru8 | 3 | 0b | Defines whether address bits [10:8] are masked.<br>1 = Masked<br>0 = No mask ("masked" means bits 10:8 are don't care) |
| MaskBits11thru8 | 4 | 0b | Defines whether address bits [11:8] are masked.<br>1 = Masked<br>0 = No mask |
| MaskBits12thru8 | 5 | 0b | Defines whether address bits [12:8] are masked.<br>1 = Masked<br>0 = No mask |
| MaskBits13thru8 | 6 | 0b | Defines whether address bits [13:8] are masked.<br>1 = Masked<br>0 = No mask |
| TMKBC_BaseAddrLow | 31:7 | 0000000h | This register defines the lower 32 bit memory address used for the TMKBC function. |
| There are actually four sets of such mapping (each set consists of register 84h, 88h, and 8Ch). The selection is controlled by PCI register 7Ch, bits [6:5]. 00 = set 0, 01 = set 1, 10 = set 2, 11 = set 3. Register 7C[6:5] SHOULD BE PROGRAMMED FIRST BEFORE 84h, 88h, AND 8Ch ARE ACCESSED. | | | |

| TMKBC_BaseAddrHigh Register- RW - 32 bits - [PCI_Reg: 88h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TMKBC_BaseAddrHigh | 31:0 | 00000000h | This register defines the upper 32 bit memory address used for the TMKBC function. This register has no meaning if bit 2 of 84h is set to 0.<br>There are actually four sets of such mapping. The selection is controlled by PCI register 7Ch, bits [6:5].<br>00 = Set 0<br>01 = Set 1<br>10 = Set 2<br>11 = Set 3 |
| There are actually four sets of such mapping (each set consists of register 84h, 88h, and 8Ch). The selection is controlled by PCI register 7Ch, bits [6:5]. 00 = Set 0, 01 = Set 1, 10 = Set 2, 11 = Set 3. Register 7C[6:5] SHOULD BE PROGRAMMED FIRST BEFORE 84h, 88h, AND 8Ch ARE ACCESSED. | | | |

| TMKBC_Remap Register- RW - 16 bits - [PCI_Reg: 8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 7:0 | 00h | |

| TMKBC_Remap Register- RW - 16 bits - [PCI_Reg: 8Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| TMKBC_Remap | 15:8 | 00h | This register defines the remap address [15:8] on the LPC bus. There are actually four sets of such mapping. The selection is controlled by PCI register 7Ch, bits [6:5].<br>00 = Set 0<br>01 = Set 1<br>10 = Set 2<br>11 = Set 3 |
| There are actually four sets of such mapping (each set consists of register 84h, 88h, and 8Ch). The selection is controlled by PCI register 7Ch, bits [6:5]. 00 = Set 0, 01 = Set 1, 10 = Set 2, 11 = Set 3. Register 7C[6:5] SHOULD BE PROGRAMMED FIRST BEFORE 84h, 88h, AND 8Ch ARE ACCESSED. | | | |

| Wide IO 2 Register- RW - 16 bits - [PCI_Reg: 90h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Base Address 2 | 15:0 | 0000h | 16-bit PCI IO base address for wide generic port range. 512 byte wide range. This function is enabled by PCI register 4Bh, bit 1 |

| SPI Base_Addr Register- RW - 16 bits - [PCI_Reg: A0h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 4:0 | 0h | |
| SPI_BaseAddr | 31:5 | 0000000h | This register defines the base address for the SPI ROM controller |

### *3.1.3 SPI ROM Controller Registers*

SPI ROM interface is a new feature added to SB600. SB600 can use SPI ROM as the BIOS ROM as well as sharing the same physical ROM with a MAC device. In addition, SB600 can act as a SPI-LPC bridge to the MAC device. In other words, the LAN data can be stored in the LPC/FWH ROM but the MAC can access it through the SPI interface.

Software can communicate with the SPI ROM through the default memory or alternate program method.

Memory access to the BIOS ROM address space is automatically handled by the hardware. The SPI ROM controller will translate the memory address onto the SPI bus and access the SPI ROM data. Any other commands besides memory_read or memory_write to the SPI ROM will need to go through the alternate program method. In this method, software will need to program the OpCode, SpiAddress, TxByteCount, RxByteCount, put the data into the transmit FIFO, and then execute the command. The hardware will then communicate with the SPI ROM using these parameters. This method is needed for the case of erasing the SPI ROM since SPI ROM has to be erased first before any new data can be written with a separate memory write command. This is also useful for querying the SPI status and vendor_ID register.

| Register Name | Offset Address |
|---|---|
| SPI_Cntrl0 | 00h |
| RestrictedCmd1 | 04h |
| RestrictedCmd2 | 08h |
| SPI_Cntrl1 | 0Ch |
| SPI_CmdValue0 | 10h |
| SPI_CmdValue1 | 14h |
| SPI_CmdValue2 | 18h |

| Register Name | Offset Address |
|---|---|
| SPI_FakeID | 1Ch |

| SPI_Cntrl0 Register- RW - 32 bits - [Mem_Reg 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPI_OpCode | 7:0 | 00h | When software uses the alternate program method to communicate with the SPI ROM, this register contains the OPCODE |
| TxByteCount | 11:8 | 0h | Number of bytes to be sent to SPI ROM |
| RxByteCount | 15:12 | 0h | Number of bytes to be received from the SPI ROM |
| ExecuteOpCode | 16 | 0b | Write 1 to execute the transaction in the alternate program registers. Write 0 has no effect. When the transaction is complete, this bit will return 0. If the command is an illegal command, the bit cannot be set and thereby cannot execute |
| Reserved | 18:17 | 00b | |
| SpiArbEnable | 19 | 1b | If a MAC is sharing the ROM with the SB, both chips will need to go through an arbitration process before either one can access the ROM. This bit enables the arbitration. If MAC is not sharing the SPI ROM, BIOS should set this bit to 0 to speed up the SPI ROM access |
| FifoPtrClr | 20 | 0b | (write only) A write of 1 to this bit will clear the internal FIFO pointer |
| FifoPtrInc | 21 | 0b | (write only ) A write of 1 to this bit will cause the internal FIFO pointer to be incremented by 1 |
| SpiAccessMacRomEn | 22 | 1b | This is a clear-once protection bit; once set, software cannot access MAC's portion of the ROM space (lower 512KB). |
| SpiHostAccessRomEn | 23 | 1b | This is a clear-once protection bit; once set, MAC cannot access BIOS ROM space (upper 512KB) |
| ArbWaitCount | 26:24 | 100b | Under ROM sharing mode (with the MAC) this defines the amount of wait time this controller will assert HOLD# before it should access the SPI ROM. This time is to allow the MAC to sample HOLD#. |
| SpiBridgeDisable | 27 | 0b | Setting this bit will disable the SPI bridge mode (SB600 will act as a SPI-LPC bridge to the MAC) |
| Reserved | 31:28 | 0h | |

| SPI_RestrictedCmd1 Register- RW - 32 bits - [Mem_Reg 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RestrictedCmd0 | 7:0 | 00h | This defines a restricted command issued by the MAC which will be checked by the SB600. If the opcode issued by the MAC matches with this register and the address space is in the BIOS space, this controller will simply ignore the command for the case of bridge mode. For peer mode, the SPI controller will jam the entire interface as an attempt to stop that transaction. Note either SpiAccessMacRomEn and/or SpiHostAccessRomEn bit is cleared, these registers become read only and cannot be changed any more. |
| RestrictedCmd1 | 15:8 | 00h | Same as RestrictedCmd0 |
| RestrictedCmd2 | 23:16 | 00h | Same as RestrictedCmd0 |
| RestrictedCmd3 | 31:24 | 00h | Same as RestrictedCmd0 |

| SPI_RestrictedCmd2 Register- RW - 32 bits - [Mem_Reg 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RestrictedCmd4 | 7:0 | 00h | Same as RestrictedCmd0 |
| RestrictedCmdWoAddr0 | 15:8 | 00h | Same as RestrictedCmd0 except this command does not have address |

| SPI_RestrictedCmd2 Register- RW - 32 bits - [Mem_Reg 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RestrictedCmdWoAddr1 | 23:16 | 00h | Same as RestrictedCmd0 except this command does not have address |
| RestrictedCmdWoAddr2 | 31:24 | 00h | Same as RestrictedCmd0 except this command does not have address |
| Note: For these registers either the SpiAccessMacRomEn and/or the SpiHostAccessRomEn bit is cleared; RestrictedCmdWoAddr1 and RestrictedCmdWoAddr2, become read only and cannot be changed. | | | |

| SPI_Cntrl1 Register- RW - 32 bits - [Mem_Reg 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPIParameters | 7:0 | 00h | This is the TX/RX FIFO port which can take up to 8 bytes. To send data to SPI ROM, software writes data into this port. To retrieve data that are received from the SPI ROM, software reads from this port. |
| FifoPtr | 10:8 | 000b | This three bits show the internal pointer location |
| TrackMacLockEn | 11 | 0b | When set, the controller will lock the SPI for the MAC when it has detected a command (from the MAC) matching the value defined in offset 10h or 11h. Conversely, it will unlock the bus when it has detected a command (from the MAC) matching the value defined in offset 12h or 13h |
| NormSpeed | 13:12 | 11b | This defines the clock speed for the non-fast read command<br>00 – Reserved<br>01 – 33Mhz<br>10 – 22 Mhz<br>11 – 16.5Mhz |
| FastSpeed[1:0] | 15:14 | 01b | This defines the clock speed for the fast speed read.<br>00 – Reserved<br>01 – 33Mhz<br>10 – 22 Mhz<br>11 – 16.5Mhz |
| WaitClkInterval | 21:16 | 22h | Timing parameters used for SPI sharing protocol |
| SetLockCmd | 22 | 0b | Lock the SPI bus on the next transaction |
| SetUnlockCmd | 23 | 0b | Unlock the SPI bus on the next transaction |
| ByteProgramCmd | 31:24 | 00h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the BYTE PROGRAM command. |

| SPI_CmdValue0 Register- RW - 32 bits - [Mem_Reg 10h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MacLockCmd0 | 7:0 | 06h | This is used to compare against the opcode sent out by the MAC. If SPI_Cntrl1[11] is set, the controller will lock the SPI bus for the MAC. In other words, the MAC has the exclusive access to the ROM; access by the CPU will be delayed until this is unlocked. This is to allow the MAC to do certain sequence of operations without interruption. |
| MacLockCmd1 | 15:8 | 20h | Same as MacLockCmd0 |
| MacUnlockCmd0 | 23:16 | 04h | This is used to compare against the opcode sent out by the MAC. If SPI_Cntrl1[11] is set, the controller will unlock the SPI bus for the MAC. In other words, access by the CPU will be allowed again. |
| MacUnlockCmd1 | 31:24 | 04h | Same as MacUnlockCmd0 |
| Note Either the SpiAccessMacRomEn and/or the SpiHostAccessRomEn bit is cleared. All of these registers become read only and cannot be changed. | | | |

| SPI_CmdValue1 Register- RW - 32 bits - [Mem_Reg 14h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| WREN | 7:0 | 06h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the WREN (write enable) command from the MAC. In the bridge mode, SB600 will need to decode commands from the MAC |
| WRDI | 15:8 | 04h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the WRDI (write disable) command from the MAC. |
| RDID | 23:16 | 9Fh | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the RDID (read ID) command from the MAC |
| RDSR | 31:24 | 05h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the RDSR (read status register) command from the MAC |
| Note: Either the SpiAccessMacRomEn and/or the SpiHostAccessRomEn bit is cleared. All of these registers become read only and cannot be changed. | | | |

| SPI_CmdValue2 Register- RW - 32 bits - [Mem_Reg 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Read | 7:0 | 03h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the Read (Read byte) command from the MAC. In the bridge mode, SB600 will need to decode commands from the MAC |
| FRead | 15:8 | 0Bh | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the FRead (fast read) command from the MAC. |
| PAGEWR | 23:16 | 0Ah | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the PAGEWR (page write) command from the MAC |
| BYTEWR | 31:24 | 02h | This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the BYTEWR (byte write) command from the MAC. |
| Note: Either the SpiAccessMacRomEn and/or the SpiHostAccessRomEn bit is cleared. All of these registers become read only and cannot be changed. | | | |

| SPI_FakeID Register- RW - 8 bits - [Offset 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SPI_FakeID | 7:0 | FFh | This is used as the faked ID value to be returned to the MAC as a response to the RDID command. This is only used under the bridge mode. |

### *3.1.4 Features of the LPC Block*

**Bus Speed:** LPC bus—33MHz

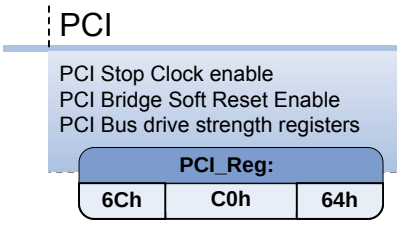
**Supported peripherals and address:**

- I/O address devices:
  - Parallel Ports: 378~37fh, 778~77fh, 278~27fh, 678~67fh, 3bc~3bfh, 7bc~7bfh
  - Serial Ports: 3f8~3ffh, 2f8~2ffh, 220~227h, 228~22fh, 238~23fh, 2e8~2efh, 338~33fh, 3e8~3efh
  - Audio: 220~233h, 240~253h, 260~273h, 280~293h
  - MIDI: 300~301h, 310~311h, 320~321h, 330~331h
  - MSS: 530~537h, 604~60bh, e80~e87h, f40~f47h
  - FDC: 3f0~3f7h (excluding 3f6h), 370~377h (excluding 376h)
  - Game Ports: 200~20fh
  - Wide Generic: Can be mapped anywhere in lower 64KB I/O address. 512 bytes wide.
  - RTC: 70~73h
  - Key Board Controller: 60h, 64h
  - ACPI Micro-Controller: 62h, 66h
  - Ad-Lib: 388h~389h
  - Super I/O Configuration: 2e~2fh
  - Alternative Super I/O Configuration: 4e~4fh

- Memory address devices:
  - ROM: supported address is in the range of 0000_0000h~ffff_ffffh
  - Firmware Hub Rom: supported address is in one of these two ranges: 000e_0000h~000f_ffffh, or ffb0_0000~ffff_ffffh

## 3.2 Host PCI Bridge Registers (Device 20, Function 4)

Note: Some PCI functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these PCI functions and the associated registers.




PCI Bridge (PCIB) has one set of configuration registers in PCI configuration space identified by PCI function 4 on the South Bridge.

**Table 3-1 PCI-to-PCI Bridge Configuration Registers Summary**

| Register Name | Offset Address |
|---|---|
| Vendor ID | 00h |
| Device ID | 02h |
| PCI Command | 04h |
| PCI Device Status | 06h |
| Revision ID/Class Code | 08h |
| Cache Line Size | 0Ch |
| Latency Timer | 0Dh |
| Header Type | 0Eh |
| Reserved | 0Fh |
| Primary Bus Number | 18h |
| Secondary Bus Number | 19h |
| Subordinate Bus Number | 1Ah |
| Secondary Latency Timer | 1Bh |
| IO Base | 1Ch |
| IO Limit | 1Dh |
| Secondary Status | 1Eh |
| Memory Base | 20h |
| Memory Limit | 22h |
| Prefetchable Memory Base | 24h |
| Prefetchable Memory Limit | 26h |
| IO Base Upper 16 Bits | 30h |
| IO Limit Upper 16 Bits | 32h |
| Capability pointer | 34h |
| Reserved | 36h |
| Interrupt Line | 3Ch |
| Interrupt Pin | 3Dh |
| Bridge Control | 3Eh |
| Chip Control | 40h |
| Diagnostic Control | 41h |
| CLK Control | 42h |
| Arbiter Control and Priority Bits | 43h |
| SMLT Performance | 44h |
| PMLT Performance | 46h |
| PCDMA | 48h |

| Register Name | Offset Address |
|---|---|
| Additional Priority | 49h |
| PCICLK Enable Bits | 4Ah |
| Misc Control | 4Bh |
| AutoClockRun Control | 4Ch |
| Dual Address Cycle Enable and PCIB_SCLK_Stop Override | 50h |
| MSI Mapping Capability | 54h |
| Signature Register for Microsoft Rework for Subtractive Decode | 58h |
| Prefetch Timeout Limit | 5Ch |
| SPCI IDSEL MaskB | 5Eh |
| Prefetch Size Control | 60h |
| Misc Control | 64h |

| Vender ID - R - 16 bits - [PCI_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| VID | 15:0 | 1002h | Vendor ID |
| Vendor ID register | | | |

| Device ID - R - 16 bits - [PCI_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DID | 15:0 | 4384h | Device ID |
| Device ID register | | | |

| Command - RW - 16 bits - [PCI_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO Enable | 0 | 0b | IO response enable,<br>0: Disabled,<br>1: Enabled,<br>PCIB responses IO space accesses on primary bus. |
| Memory Enable | 1 | 0b | Memory response enable, '0' disabled, '1' enabled, PCIB responses memory space accesses on primary bus. |
| Master Enable | 2 | 0b | Master enable, '0' disabled, '1' enabled, the ability of PCIB to act as a PCI bus master on primary bus. |
| Special Enable | 3 | 0b | Hardwired to 0 to indicate that PCIB ignores special cycles. |
| Mem Invalidate | 4 | 0b | Hardwired to 0 to indicate that PCIB doesn't issue memory write and invalidate command by itself. |
| VGA Snoop Enable | 5 | 0b | VGA snoop enable, when '0' VGA palette write transactions on the primary interface are ignored unless it falls into PCIB's IO address range. When '1' VGA palette write transactions are positively decoded and forwarded downstream. |
| Parity Error Enable | 6 | 0b | Parity Error Response, '0' disables PCIB to assert P_SERR# and P_PERR# or report Detected Parity Error to the status register, '1' enables PCIB. |
| Addr Stepping Enable | 7 | 0b | Controls whether or not to do address/data stepping, PCIB doesn't. Read Only |
| System Error Enable | 8 | 0b | SERR# enable, '0' disables PCIB to assert P_SERR# and report Signaled System Error bit, '1' enables PCIB. |
| Fast Back-to-Back Enable | 9 | 0b | Hardwired to 0 to indicate that PCIB is not capable of issuing fast back-to-back transactions on the primary bus. |
| Reserved | 15:10 | 00h | Reserved |
| PCI Command register | | | |

| Status- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | Reserved |

| Status- RW - 16 bits - [PCI_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities List | 4 | 0b | Read only. This bit is 1 when Offset 40h [3] = 1. At other time this bit is 0.<br><br>0 – Bridge does not support the Capabilities List<br>1 – Bridge supports the Capabilities List (Offset 34h is the pointer to the data structure). |
| 66MHz Capable | 5 | 1b | Hardwired to 1 to indicate PCIB support of 66MHz primary interface. |
| Reserved | 6 | 0b | Reserved |
| Fast Back-to-Back Capable | 7 | 1b | Hardwired to 1 to indicate PCIB is capable of accepting fast back-to-back transactions on the primary bus. |
| Master Parity Error | 8 | 0b | Master Data Parity Error, assertion of P_PERR# (when PCIB acts as a master) is received, write clears it. |
| DevSel Timing | 9:10 | 01b | Hardwired to 01b to indicate PCIB will assert DEVSEL# with medium timing. |
| Target Abort | 11 | 0b | Signaled Target Abort, write clears it. |
| Received Target Abort | 12 | 0b | Received Target Abort, write clears it. |
| Received Master Abort | 13 | 0b | Received Master Abort, write clears it. |
| Master Abort | 14 | 0b | Signaled System Error bit, write clears it. |
| Parity Error | 15 | 0b | Detected Parity Error, PCIB detected a parity error and will assert P_PERR#, write clears it. |
| PCI device status register. | | | |

| Revision ID/Class Code- R - 32 bits - [PCI_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Revision ID | 7:0 | 00h | These bits are hardwired to 00h to indicate the revision level of the chip design (for the SB600). |
| Class Code | 31:8 | 060401h/<br>060400h | A class code of 06h indicates a bridge device, a subclass code of 04h indicates PCI bridge and a programming interface of '01h' indicates subtractive decoding on primary bus is supported. Based on Reg0x4B[7], this register is programmed as positive decode bridge or subtractive decode bridge. |
| Revision ID/Class Code register. | | | |

| Cache Line Size- RW - 8 bits - [PCI_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Cache Line Size | 7:0 | 00h | Read Only |
| Cache line size register | | | |

| Primary Master Latency Timer- RW - 8 bits - [PCI_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Prim Latency Timer | 7:0 | 00h | Primary master latency timer. Sets the minimum time that the Primary bus master can retain the ownership of the bus. |
| Primary Master Latency timer register | | | |

| Header Type- R - 8 bits - [PCI_Reg: 0Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Header Type | 7:0 | 81h | Indicates the bridge is a multi-function device. |
| Header type register | | | |

| Primary Bus Number- RW - 8 bits - [PCI_Reg: 18h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Primary Bus Number | 7:0 | 00h | Bus number of the PCI bus to which the primary interface is connected. |
| Primary Bus Number register | | | |

| Secondary Bus Number - RW - 8 bits - [PCI_Reg: 19h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Secondary Bus Number | 7:0 | 00h | Bus number of the PCI bus to which the secondary interface is connected. |
| Secondary Bus Number register | | | |

| SUBBN- RW - 8 bits - [PCI_Reg: 1Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SubordinateBusNum | 7:0 | 00h | Bus number of the highest numbered PCI bus behind PCIB. |
| Subordinate Bus Number register | | | |

| Secondary Latency Timber- RW - 8 bits - [PCI_Reg: 1Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Secondary Latency Timer | 7:0 | 00h | Secondary Master latency control timer. Sets the minimum time that the Secondary bus master can retain the ownership of the bus. |
| Secondary Master Latency Timer register | | | |

| IO Base- RW - 8 bits - [PCI_Reg: 1Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO16 | 1:0 | 00b | Indicates a 16-bit IO address space. Read Only<br>Can be changed to 32-bit when bit[29] of regx48 is set |
| Reserved | 3:2 | 00b | Reserved |
| IOBase | 7:4 | 0h | Defines the bits [15:12] of the base address of 16-bit or 32-bit IO space. |
| IO base register | | | |

| IO Limit- RW - 16 bits - [PCI_Reg: 1Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO16 | 1:0 | 00b | Indicates a 16-bit IO address space. Read Only<br>Can be changed to 32-bit when bit[29] of regx48 is set |
| Reserved | 3:2 | 00b | Reserved |
| IO Limit | 7:4 | 0h | Defines bits [15:12] of the limit of 16-bit or 32-bit IO space. |
| IO Limit register | | | |

| Secondary Status- RW - 16 bits - [PCI_Reg: 1Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 4:0 | 00h | Reserved |
| Sec 66MHz Capable | 5 | 0b | Indicates PCIB doesn't support 66MHz secondary interface. Read Only. |
| Reserved | 6 | 0b | Reserved |
| Secondary Fast Back-to-Back Capable | 7 | 1b | Indicates PCIB is capable of accepting fast back-to-back transactions on the secondary bus. Read Only. |
| Secondary Master Data Parity Error | 8 | 0b | Master Data Parity Error on the secondary bus, assertion of S_PERR# (when PCIB acts as a master) is received, write clears it. |
| Secondary DevSel Timing | 10:9 | 01b | DEVSEL# timing, indicates PCIB will assert DEVSEL# with medium timing on the secondary bus. Read Only. |
| Secondary Target Abort | 11 | 0b | Signaled Target Abort on the secondary bus, write clears it. |

| Secondary Status- RW - 16 bits - [PCI_Reg: 1Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Received Secondary Target Abort | 12 | 0b | Received Target Abort on the secondary bus, write clears it. |
| Received Secondary Master Abort | 13 | 0b | Received Master Abort on the secondary bus, write clears it. |
| Received Serr | 14 | 0b | Received System Error on the secondary bus, PCIB asserts P_SERR# to propagate the error back to the primary bus, write clears it. |
| Data Parity Error | 15 | 0b | Detected Parity Error on the secondary bus, PCIB detected a parity error and will assert S_PERR#, write clears it. |
| Secondary status register | | | |

| Memory Base- RW - 16 bits - [PCI_Reg: 20h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | Indicates a non-prefetchable 32-bit memory space. Read Only |
| Non Pref Mem Base | 15:4 | 000h | Defines the highest 12 bits ([31:20]) of the base address of this 32-bit memory space. |
| Memory base register | | | |

| Memory Limit- RW - 16 bits - [PCI_Reg: 22h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | Indicates a non-prefetchable 32-bit memory space. Read Only. |
| Non Pref Mem Limit | 15:4 | 000h | Defines the highest 12 bits ([31:20]) of the upper limit of this 32-bit memory space. |
| Memory limit register | | | |

| Prefetchable Memory Base- RW - 16 bits - [PCI_Reg: 24h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | Indicates a 32-bit only +memory space. Read Only |
| Pref Mem Base | 15:4 | 000h | Defines the highest 12 bits ([31:20]) of the base address of this 32-bit memory space. |
| Prefetchable memory base register | | | |

| Prefetchable Memory Limit- RW - 16 bits - [PCI_Reg: 26h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | Indicates a 32-bit only memory space. Read Only. |
| Perf Mem Limit | 15:4 | 000h | Defines the highest 12 bits ([31:20]) of the upper limit of this 32-bit memory space. |
| Prefetchable memory limit register | | | |

| IO Base Upper 16 Bits- RW - 16 bits - [PCI_Reg: 30h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IOBase Upper | 15:0 | 0000h | Top 16 bits of the base address of 32-bit IO transactions. If the IO address decode mode bit (Regx48 bit[29] ) is clear then these bits will be zero |
| IO base upper 16 bits | | | |

| IO Limit Upper 16 bits- RW - 16 bits - [PCI_Reg: 32h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IOLimit Upper | 15:0 | 0000h | Top 16 bits of the upper limit of 32-bit IO transactions. If the IO address decode mode bit (Regx48 bit[29] ) is clear then these bits will be zero |

| IO Limit Upper 16 bits- RW - 16 bits - [PCI_Reg: 32h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| IO limit upper 16 bits | | | |

| Capabilities Pointer- R - 8 bits - [PCI_Reg: 34h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Capabilities Pointer | 7:0 | 00h | Enhanced Capability Pointer. Read Only.<br>Value = 54h when Offset 40h [3] MSI Guide Bit set to '1' |
| Capability Pointer register | | | |

| Interrupt Line- R - 8 bits - [PCI_Reg: 3Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Line | 7:0 | 00h | Interrupt pin routing information, used as communication window between BIOS and the device driver. |
| Interrupt Line register | | | |

| Interrupt Pin- R - 8 bits - [PCI_Reg: 3Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Interrupt Pin | 7:0 | 00h | Interrupt pin usage information. '0' indicates PCIB not supporting interrupt routing. |
| Interrupt Pin register | | | |

| Bridge Control- RW - 16 bits - [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Parity Error Enable | 0 | 0b | Parity Error Response, '0' disables PCIB to assert P_SERR# and S_PERR# or report Detected Parity Error to the secondary status register, '1' enables PCIB. |
| Serr# Enable | 1 | 0b | SERR# forward enable, when '0' PCIB doesn't drive P_SERR# when it detects S_SERR#. |
| ISA_Enable | 2 | 0b | ISA enable, when '0' no ISA address mode, when '1' ISA address mode is supported. |
| VGA_Enable | 3 | 0b | VGA enable, '0' disabled, '1' enabled. |
| VGA 16-bit decode | 4 | 0b | This bit only has meaning if either bit 3 (VGA Enable) of this register, or bit 5 (VGA Palette Snoop Enable) of the Command Register, is also set to 1, thereby enabling VGA I/O decoding and forwarding by the bridge.<br>The status after reset is 0.<br>This read/write bit enables system configuration software to select between 10- and 16-bit I/O address decoding for all VGA I/O register accesses that are forwarded from primary to secondary.<br><br>0 - Execute 10-bit address decodes on VGA I/O accesses.<br>1 - Execute 16-bit address decodes on VGA I/O accesses |
| Master Abort Report | 5 | 0b | Master abort mode, '0' does not report master aborts (return FFFF,FFFFh on reads and discard data on write), '1' reports master aborts by signaling target abort or by asserting SERR# if enabled. |
| Secondary Reset | 6 | 0b | Secondary bus reset, '0' disabled, '1' trigger reset.. This bit can be masked using ACPI PCI config register x6c bit 5 |
| Secondary Fast Back-to-Back Enable | 7 | 0b | Secondary bus fast back-to-back enable. PCIB is not capable of issuing fast back-to-back transactions on the secondary bus. Read Only |
| Primary Discard Timer | 8 | 0b | Primary Discard Timer configuration, '0' configures the timer to 15-bit, '1' configures the timer to 10-bit. |

| Bridge Control- RW - 16 bits - [PCI_Reg: 3Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Secondary Discard Timer | 9 | 0b | Secondary Discard Timer configuration, '0' configures the timer to 15-bit, '1' configures the timer to 10-bit. |
| Discard Timer Status | 10 | 0b | Discard Timer Status, '0' no discard timer error, '1' discard timer error. |
| Discard Timer Serr# Enable | 11 | 0b | Discard Timer SERR# enable, '0' disabled, '1' enabled. |
| Reserved | 15:12 | 0h | Reserved |
| Bridge control register | | | |

| CPCTRL- RW - 8 bits - [PCI_Reg: 40h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | Reserved |
| Mem Write Size Ctrl | 1 | 0b | Controls the Memory write size. When set the memory write size will be cacheline aligned else it will be 32 byte aligned. |
| Lock Enable | 2 | 1b | Downstream locked transaction enable. |
| MSIC ap Enable | 3 | 0b | MSI Capability Enable Guide bit. '1' enabled, '0' disabled. Setting this bit to '1' will change the status of Offset 04h [20] from '0' to '1' |
| Prefetch Disable for Upstream Mem Read | 4 | 0b | In A11~A13:<br>0: Prefetch is enabled for upstream normal memory read (other than read line or read multiple). However, if reg0x64[7]=0, then this prefetch is still disabled.<br>1: Prefetch is disabled for upstream normal memory read (other than read line or read multiple).<br><br>In A21 and later versions: Reserved.<br><br>Also refer to reg0x64[7] and reg0x64[21]. |
| Sub Decode Enable | 5 | 0b | This bit is used only when reg0x4B[7] = 1.<br>1: Subtractive decoding is enabled.<br>0: Subtractive decoding is disabled. |
| Bridge Lock State | 7:6 | 00b | Bridge secondary master lock states. Read Only<br>'00' – FREE<br>'01' – BUSY<br>'10' – REQ<br>'11' – LOCKED |
| Chip control register | | | |

| DCTRL- RW - 8 bits - [PCI_Reg: 41h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | Reserved |
| Timer Test Mode | 2:1 | 00b | Timer Test Mode,<br>'00' – normal operation, all bits are exercised.<br>'01' – Byte 1 is exercised.<br>'10' – Byte 2 is exercised.<br>'11' – Byte 0 is exercised. |
| Force Parity Error | 3 | 0b | Force receiving/generating parity error, '0' receiving, '1' generating. |
| Force Data Parity Error | 4 | 0b | Force data parity error, '0' disabled, '1' enabled. |
| Force Addr Parity Error | 5 | 0b | Force address parity error, '0' disabled, '1' enabled. |
| Force Sec Parity Error | 6 | 0b | Force secondary Parity Error Mode, '0' disabled, '1' enabled. |
| Force Prim Parity Error | 7 | 0b | Force primary Parity Error Mode, '0' disabled, '1' enabled. |
| Diagnostic control register | | | |

| CLKCTRL- RW - 8 bits - [PCI_Reg: 42h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCICLKStopEnable | 0 | 0b | 33MHz PCICLKs request bit; when '1,' 33 MHz PCI Clocks are requested to stop. |
| PCICLKStopStatus | 1 | 0b | Read only.<br>33MHz PCICLKs stop status: '1' stopped, '0' running. |
| PCICLK0Enable | 2 | 1b | 33MHz PCICLK0 enable. |
| PCICLK1Enable | 3 | 1b | 33MHz PCICLK1 enable. |
| PCICLK2Enable | 4 | 1b | 33MHz PCICLK2 enable. |
| PCICLK3Enable | 5 | 1b | 33MHz PCICLK3 enable. |
| P2SControl | 6 | 0b | P_CLK domain to S_CLK domain synch-up disable. |
| S2PControl | 7 | 0b | S_CLK domain to P_CLK domain synch-up disable. |
| Clock control register | | | |

| ARCTRL- RW - 8 bits - [PCI_Reg: 43h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 6:0 | ffh | Reserved |
| ArbiterEnable | 7 | 1b | Arbiter enable. '0' disabled to give PCIB the exclusive ownership of the secondary bus. |
| Arbiter control register | | | |

| SMLT_PERF- RW - 16 bits - [PCI_Reg: 44h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| SMLT_Perf | 15:0 | 0000h | Count the total number of a burst being broken into multiple transactions due to MLT timeout. |
| Secondary MLT performance register | | | |

| PMLT_PERF- RW - 16 bits - [PCI_Reg: 46h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PMLT_Perf | 15:0 | 0000h | Count the total number of a burst being broken into multiple transactions due to MLT timeout. |
| Primary MLT performance register | | | |

| PCDMA- RW - 8 bits - [PCI_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCDMA Device Enable A | 0 | 0b | Device enable for request 3. Needs to be enabled when there is a PCDMA device corresponding to request 3 |
| PCDMA Device Enable B | 1 | 0b | Device enable for request 4. Heeds to be enabled when there is a PCDMA device corresponding to request 4 |
| Fast Back to Back Retry Enable | 2 | 0b | Retry Fast Back to Back transactions on Write buffer full. |
| Lock Operation Enable | 3 | 1b | When reg0x40[2]=1, this bit should be set to 1 for the proper operation of the PCI LOCK# function. |
| Reserved | 7:4 | 00h | Reserved |
| PCDMA device Enable bits | | | |

| Additional Priority- Bits RW - 8 bits - [PCI_Reg: 49h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| PCDMA Priority | 1 | 0b | If enabled includes PCDMA request into high priority list |
| Reserved | 7:2 | 00h | Reserved |
| Priority Bits | | | |

| PCICLK Enable Bits- RW - 8 bits - [PCI_Reg: 4Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCICLK4Enable | 0 | 1b | 33MHz PCICLK4 enable. |
| PCICLK5Enable | 1 | 1b | 33MHz PCICLK 5 enable. |
| PCICLK6Enable | 2 | 1b | 33MHz PCICLK 6 enable. |
| PCICLK7Enable | 3 | 1b | 33MHz PCICLK 7 enable. |
| Reserved | 7:4 | 3h | Reserved |
| PCICLK Enable bits | | | |

| Misc Control RW - 8 bits - [PCI_Reg: 4Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| GNT Bus Idle check enable | 0 | 0b | When enabled, the PCI arbiter checks for the Bus Idle before asserting GNT#. |
| Memory Read Burst Size | 4:1 | 0h | Specifies up to how many double words burst to support during an upstream or downstream memory read.<br>[4:1] =<br>1xxx: Burst up to 16 double words<br>01xx: Burst up to 8 double words<br>001x: Burst up to 4 double words<br>0001: Burst up to 2 double words<br>Others: Burst up to 8 double words<br><br>Note 1: It has no effect on a downstream normal memory read (other than read line and read multiple), which has no burst in this design.<br>Note 2: It has no effective on an upstream memory read if the read is prefetchable as specified by reg0x64[7], reg0x64[21], and reg0x40[4], because a prefetchable read can have unlimited burst. |
| IOMode | 5 | 0b | Control bit to change the IO addressing mode to 32/16 bit.<br>0 – 16 bits;<br>1 – 32 bit. |
| MemReadCmdMatch | 6 | 0b | Control bit to enable the match of memory read/memory read line commands when there is a read command in the delay queue. |
| SubDecodeEnable | 7 | 0b | Control bit for the subtractive decode status (09h).<br>0 – No subtractive decode;<br>1 – Whether the subtractive decode is enabled depends on reg0x40[5]. |
| Misc control Register | | | |

| AutoClockRun control RW - 32 bits - [PCI_Reg: 4Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Autoclkrun Enable | 0 | 0b | Enables the auto clkrun functionality |
| Autoclkrun count | 31:1 | 0000_000 0h | Number of cycles after which the secondary clock stops when clkrun is enabled |
| Auto ClockRun control register | | | |

| Dual Address Cycle Enable and PCIB_CLK_Stop Override - RW - 16 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCIB_Dual_EN_up | 0 | 0b | Enables decoding of Dual Address Cycle on secondary side for upstream memory transactions |
| PCIB_Dual_EN_dn | 1 | 0b | Enables decoding of Dual Address Cycle on secondary side for downstream memory transactions |
| Reserved | 5:2 | 0h | |

| Dual Address Cycle Enable and PCIB_CLK_Stop Override - RW - 16 bits - [PCI_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ClkrunOvrridePCICLK | 6 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK continues to run. |
| ClkrunOvrridePCICLK1 | 7 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK1 continues to run. |
| ClkrunOvrridePCICLK2 | 8 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK2 continues to run. |
| ClkrunOvrridePCICLK3 | 9 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK3 continues to run. |
| ClkrunOvrridePCICLK4 | 10 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK4 continues to run. |
| ClkrunOvrrideLPCCLK | 11 | 0b | When set, overrides the CLKRUN# and LPCCLK continues to run. |
| ClkrunOvrrideLPCCLK1 | 12 | 0b | When set, overrides the CLKRUN# and LPCCLK1 continues to run |
| ClkrunOvrridePCICLK7 | 13 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK7 continues to run. |
| ClkrunOvrridePCICLK8 | 14 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK8 continues to run. |
| ClkrunOvrridePCICLK9 | 15 | 0b | When set, overrides the CLKRUN# and 33MHz PCICLK9 continues to run. |
| Dual Address Cycle Enable and PCIB_CLK_Stop Override | | | |

| MSI Mapping Capability – R - 32 bits - [PCI_Reg: 54h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| MSI Cap ID | 7:0 | 08h | MSI Capability ID |
| MSI Cap Pointer | 15:8 | 00h | MSI Capabilities Pointer |
| MSI Cap Enable | 16 | 1b | MSI Capabilities Enable |
| MSI Fixed | 17 | 1b | MSI Fixed |
| MSI Reserved | 26:18 | 000h | Reserved |
| MSI CapType | 31:27 | 15h | MSI Capability Type |
| MSI Mapping Capability | | | |

| Signature Register for Microsoft Rework for Subtractive Decode - R - 32 bits - [PCI_Reg: 58h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Signature Register for Microsoft Rework for Subtractive Decode | 31:0 | 00000000h | When Microsoft Rework for Subtractive Decode is done, this register will contain the signature value |
| Signature Register for Microsoft Rework for Subtractive Decode | | | |

| Prefetch Timeout Limit - 16 bits - [PCI_Reg: 5Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Prefetch Timeout Limit | 15:0 | 0085h | When deep prefetch is enabled (reg0x64[7]=1), this timer determines when to flush the staled data in the buffer. Each count is 30ns |

| SPCI IDSEL MaskB - 16 bits - [PCI_Reg: 5Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| PCI IDSEL MaskB | 15:0 | FFFFh | Each bit represents the masking the specific device on the PCI bus. The purpose of this register is to hide the device from OS<br>0—The corresponding IDSEL bit is masked<br>1—The corresponding IDSEL bit is not masked. |

| Prefetch Size Control - 32 bits - [PCI_Reg: 60h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Read Size | 2:0 | 2h | If prefetch function is enabled, this defines the number of initial prefetch cachelines for a PCI READ command |
| Read Size Adjustment | 3 | 1b | When set, PCIBridge will adjust the prefetch size for READ automatically. If this bit is 0, then the prefetch size is always defined by bits [2:0] |
| Read Line Size | 6:4 | 4h | If prefetch function is enabled, this defines the number of initial prefetch cachelines for a PCI READLINE command |
| Read Line Adjustment | 7 | 1b | When set, PCIBridge will adjust the prefetch size for READ_LINE automatically. If this bit is 0, then the prefetch size is always defined by bits [6:4] |
| Read Multiple Size | 10:8 | 6h | If prefetch function is enabled, this defines the number of initial prefetch cachelines for a PCI READ_MULTIPLE command |
| Read Multiple Adjustment | 11 | 1b | When set, PCIBridge will adjust the prefetch size for READ_MULTIPLE automatically. If this bit is 0, then the prefetch size is always defined by bits [10:8] |
| Prefetch Size Lower Limit | 14:12 | 0h | The lower limit of the adjusted prefetch size. |
| Reserved | 15 | 0b | |
| Prefetch Size Upper Limit | 18:16 | 7h | The upper limit of the adjusted prefetch size. |
| Reserved | 19 | 0b | |
| Prefetch Size Mlt Enable | 20 | 1b | This also controls how PCIBridge adjusts the auto-prefetch size. When set, PCIBridge will only adjust the prefetch size if it knows it does not have enough or has too much data in the prefetch buffer. Recommendation is to always set this bit. |
| Reserved | 31:21 | 0h | |
| The fields in this register are effective only when prefetch is enabled (reg0x64[7]=1). | | | |

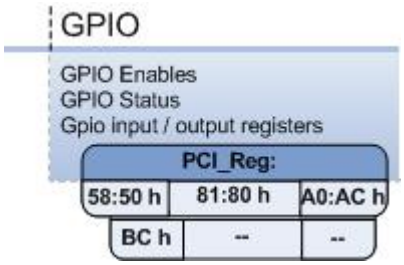
| Misc Control Register - 32 bits - [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Downstream Config Cycle Flush Enable | 0 | 0h | When this bit is 1, any downstream config cycle will flush all the upstream read prefetch buffers. |
| Downstream Write Cycle Flush Enable | 1 | 1h | When this bit is 1, any downstream non-config write cycle will flush all the upstream read prefetch buffers. |
| Downstream Read Cycle Flush Enable | 2 | 0h | When this bit is 1, any downstream non-config read cycle will flush all the upstream read prefetch buffers. |
| Prefetch Buffer Timeout Enable | 3 | 1h | When this bit is 1, upstream read prefetch buffer timeout mechanism is enabled. If data stay in a buffer longer than the time specified in Prefetch Timeout Limit (Reg5Ch), the buffer will be flushed. |
| AB Masking Prefetch Request Enable | 4 | 1h | The purpose of this is to improve the internal bus efficiency and the recommendation is to have it set to 1 |
| AB Masking Non-prefetch Request Enable | 5 | 1h | The purpose of this is to improve the internal bus efficiency and the recommendation is to have it set to 1 |
| Downstream Cycle Flush Control | 6 | 0h | 0—If a downstream cycle is qualified to flush upstream prefetch read buffer (depending on bit[2:0] in this register), the flush happens when the cycle is sent out onto PCI bus with at least one data phase..<br>1—If a downstream cycle is qualified to flush upstream prefetch read buffer (depending on bit[2:0] in this register), the flush happens as soon as the cycle arrives at PCI bus. |

| Misc Control Register - 32 bits - [PCI_Reg: 64h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Prefetch Enable For Upstream Read Line and Read Multiple | 7 | 1h | 0: Prefetch is disabled for upstream memory read line and memory read multiple.<br>1: Prefetch is enabled for upstream memory read line and memory read multiple.<br>Also refer to reg0x40[4] and reg0x64[21]. |
| PCI5 Enable | 8 | 0h | Setting this bit will enable PCIGNT5#/PCIREQ5# and PCICLK7.<br>Note: These three pins are all using this single configuration bit to enable them into PCI functionality. Care should therefore be taken to ensure that they are used as intended. Since PCICLK7 is not programmed as PCICLK by default, the BIOS needs exercise a certain sequence in order to make the clock valid. For more information consult the *AMD SB600 Register Programming Requirements* guide. |
| Arbiter 2 Enable | 9 | 0h | Enables the use of the new PCI bus arbiter to replace the old arbiter. |
| Hold Current Grant | 10 | 0h | When set, PCIGNT# will not be deasserted until the requesting agent deasserts its PCIREQ#. This only applies to the new PCI bus arbiter |
| Single Cycle Prefetch Control | 11 | 1h | If a prefetch read cycle comes from SPCI bus, prefetching may not be necessary if the cycle is single data phase. PCIB can optionally treat the single cycle as non-prefetch and only asks for one dword from AB.<br>1—Turn on the ability to treat single data phase cycle as non-prefetch cycle.<br>0—Turn off the ability to treat single data phase cycle as non-prefetch cycle. |
| Fast Grant Deassert En | 12 | 0b | This control bit applies only to the old (default) PCI arbiter. Normally PCIGNT# is deasserted two clocks after PCIREQ# deasserts. With this bit set, PCIGNT# will deassert 1 clock after PCIREQ# deasserts. Recommendation is to have this bit set. |
| Reserved | 20:13 | 0h | |
| Prefetch Disable for Upstream Mem Read | 21 | 0h | In A11~A13: Reserved.<br><br>In A21 and later versions:<br>0: Prefetch is enabled for upstream normal memory read (other than read line or read multiple). However, if reg0x64[7]=0, then this prefetch is still disabled.<br>1: Prefetch is disabled for upstream normal memory read (other than read line or read multiple).<br>Also refer to reg0x40[4] and reg0x64[7]. |
| Reserved | 31:22 | 0h | |

# 4 Register Descriptions: General Purpose Functions/Interrupt Controllers/Support Function Pins

## 4.1 GPIO/GPOC

Note: Some GPIO functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these GPIO functions and the associated registers.




### *4.1.1 GPIO*

The GPIO pins, with the exception of GPIO0/2/10, cannot generate any events. The three lines of GPIO0/2/10 can act either as GPIO lines or generate events. Refer to *section 4.2.2* on how to program them as event lines. The power-on default of the GPIO pins is GPI, but the BIOS can configure that to GPO. Hardware strap can also enable the GPIO pins as dedicated functional pins if they are multi-functional pins.

**Table 4-1: GPIO Pins**

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| GPIO0/ SSMUXSEL/ SATA_IS3# | PM IO Reg60h[Bit 7]<br>0: SSMUXSEL<br>1: GPIO if not used by SATA | Reg80h[Bit 4]<br>0: Output<br>1: Input (Tri-state) | Reg81h[Bit 0] | Reg80h[Bit 0] | S0 |
| GPIO1/ ROM_CS# | If the chip is strapped to use the ROM option on the PCI bus, this GPIO becomes ROM_CS# | Reg80h[Bit 5]<br>0: Output<br>1: Input (Tri-state) | Reg81h[Bit 1] | Reg80h[Bit 1] | S0 |
| GPIO2/ SPKR | PM IO Reg60h[Bit 5]<br>0: GPIO<br>1: SPKR | Reg80h[Bit 6]<br>0: Output<br>1: Input (Tri-state) | Reg81h[Bit 2] | Reg80h[Bit 2] | S0 |
| GPIO3/ FAN0 | PM IO Reg60h[Bit 6]<br>0: GPIO<br>1: FAN0 | Reg80h[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg81h[Bit 3] | Reg80h[Bit 3] | S0 |

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| GPIO4/<br>SMARTVOLT/<br>SATA_IS2# | SMBus Reg5Eh[Bit 7]<br>0: GPIO if not used by SATA<br>1: SMARTVOLT | RegA9h[Bit 0]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | RegAAh[Bit 0]<br>***Note 3** | RegA8h[Bit 0] | S0 |
| GPIO5/<br>SHUTDOWN#/<br>SMARTVOLT2 | SMBus Reg9Ah[Bit7]<br>1 – SMARTVOLT2<br>0 – use PMIO_59h[Bit5] to set function<br>PM IO Reg59h[Bit 5]<br>0: GPIO<br>1: SHUTDOWN# | SMBus Reg9Ah[7]=0<br>RegA9h[Bit 1]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | RegAAh[Bit 1]<br>***Note 3** | RegA8h[Bit 1] | S0 |
| GPIO6/<br>GHI#/<br>SATA_IS1# | PM IO Reg60h[Bit 7]<br>0: GHI#<br>1: GPIO if not used by SATA | RegA9h[Bit 2]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | RegAAh[Bit 2]<br>***Note 3** | RegA8h[Bit 2] | S0 |
| GPIO7/<br>WD_PWRGD | Strap on PCI AD [23]<br>0: GPIO<br>1: WD_PWRGD | RegA9h[Bit 3]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | RegAAh[Bit 3]<br>***Note 3** | RegA8h[Bit 3] | S0 |
| GPIO8/<br>DDC1_SDA | Input/Output pin only | Reg A9h[Bit 4]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | Reg AAh[Bit 4]<br>***Note 3** | Reg A8h[Bit 4] | S0 |
| GPIO9/<br>DDC1_SCL | Input/Output pin only | Reg A9h[Bit 5]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | Reg AAh[Bit 5]<br>***Note 3** | Reg A8h[Bit 5] | S0 |
| GPIO10/<br>SATA_IS0# | GPIO if not used by SATA | Reg ABh[Bit 1]<br>0: Output<br>1: Input (Tri-state) | Reg ABh[Bit 2] | Reg ABh[Bit 0] | S0 |
| GPIO11/<br>SPI_DO | SMBus RegABh[Bit 6]<br>0: SPI_DO<br>1: GPIO | Reg A9h[Bit 6]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | Reg AAh[Bit 6]<br>***Note 3** | Reg A8h[Bit 6] | S0 |
| GPIO12/<br>SPI_DI | SMBus RegABh[Bit 7]<br>0: SPI_DI<br>1: GPIO | Reg A9h[Bit 7]<br>0: Output<br>1: Input (Tri-state)<br>***Note 2** | Reg AAh[Bit 7]<br>***Note 3** | Reg A8h[Bit 7] | S0 |
| GPIO13/<br>LAN_RST# | SMBus Reg83h[Bit 4]<br>0: LAN_RST#<br>1: GPIO | Reg 82h[Bit 4]<br>0: Output<br>1: Input (Tri-state) | Reg 83h[Bit 0] | Reg 82h[Bit 0] | S0 |
| GPIO14/<br>ROM_RST# | SMBus Reg83h[Bit 5]<br>0: ROM_RST#<br>1: GPIO | Reg 82h[Bit 5]<br>0: Output<br>1: Input (Tri-state) | Reg 83h[Bit 1] | Reg 82h[Bit 1] | S0 |

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| GPIO[30:15]/ IDE_D[15:0] | SMBus Reg62h[Bit 7]<br>0: IDE_D<br>1: GPIO | RegA2h[Bit15:0]<br>0: Output<br>1: Input (Tri-state)<br>***Note 5** | RegA4h[Bit15:0] | RegA0h[Bit15:0] | S0 |
| GPIO31/ SPI_HOLD# | SMBus Reg83h[Bit 6]<br>0: SPI_HOLD#<br>1: GPIO | Reg 82h[Bit 6]<br>0: Output<br>1: Input (Tri-state) | Reg 83h[Bit 2] | Reg 82h[Bit 2] | S0 |
| GPIO32/ SPI_CS# | SMBus Reg83h[Bit 7]<br>0: SPI_CS#<br>1: GPIO | Reg 82h[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg 83h[Bit 3] | Reg 82h[Bit 3] | S0 |
| GPIO33/ INTE# | SMBus RegBDh[Bit 4]<br>0: INTE#<br>1: GPIO<br>***Note 4** | RegBCh[Bit 4]<br>0: Output<br>1: Input (Tri-State) | RegBDh[Bit 0]<br>***Note 4** | RegBCh[Bit 0] | S0 |
| GPIO34/ INTF# | SMBus RegBDh[Bit 5]<br>0: INTF#<br>1: GPIO<br>***Note 4** | RegBCh[Bit 5]<br>0: Output<br>1: Input (Tri-State) | RegBDh[Bit 1]<br>***Note 4** | RegBCh[Bit 1] | S0 |
| GPIO35/ INTG# | SMBus RegBDh[Bit 6]<br>0: INTG#<br>1: GPIO<br>***Note 4** | RegBCh[Bit 6]<br>0: Output<br>1: Input (Tri-State) | RegBDh[Bit 2]<br>***Note 4** | RegBCh[Bit 2] | S0 |
| GPIO36/ INTH# | SMBus RegBDh[Bit 7]<br>0: INTH#<br>1: GPIO<br>***Note 4** | RegBCh[Bit 7]<br>0: Output<br>1: Input (Tri-State) | RegBDh[Bit 3]<br>***Note 4** | RegBCh[Bit 3] | S0 |
| GPIO37/ DPSLP_OD# | SMBus RegA7h[Bit 4]<br>0: DPSLP_OD#<br>1: GPIO | Reg A6h[Bit 4]<br>0: Output enable<br>1: Input (Tri-state) | RegA7h[Bit 0] | RegA6h[Bit 0] | S0 |
| GPIO38/ AC_BITCLK | PM IO Reg59h[Bit 1:0]<br>00/01/10: AC97 port<br>11: GPIO | Extend Reg 05h[Bit 1]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 03h[Bit 1] | Extend Reg 07h[Bit 1] | S0 |
| GPIO39/ AC_SDOUT | PM IO Reg59h[Bit 1:0]<br>00/01/10: AC97 port<br>11: GPIO | Extend Reg 05h[Bit 3]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 03h[Bit 3] | Extend Reg 07h[Bit 3] | S0 |
| GPIO40/ AC_SYNC | PM IO Reg59h[Bit 1:0]<br>00/01/10: AC97 port<br>11: GPIO | Extend Reg 05h[Bit 2]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 03h[Bit 2] | Extend Reg 07h[Bit 2] | S0 |
| GPIO41/ SPDIF_OUT/ PCICLK7 | PCIB Reg64h[Bit 8]<br>0: GPIO or AC97 port<br>1: PCICLK7<br>PM IO Reg59h[Bit 0]<br>0: AC97 port<br>1: GPIO | Extend Reg 05h[Bit 4]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 03h[Bit 4] | Extend Reg 07h[Bit 4] | S0 |

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| GPIO42/ ACZ_SDIN0 | SMBus Extend Reg 00h[Bit1:0]<br>00/11: GPIO port<br>01: AC97 port<br>10: HD Audio port | Extend Reg 04h[Bit 0]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 02h[Bit 0] | Extend Reg 06h[Bit 0] | S0 |
| GPIO43/ ACZ_SDIN1 | SMBus Extend Reg 00h[Bit3:2]<br>00/11: GPIO port<br>01: AC97 port<br>10: HD Audio port | Extend Reg 04h[Bit 1]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 02h[Bit 1] | Extend Reg 06h[Bit 1] | S0 |
| GPIO44/ ACZ_SDIN2 | SMBus Extend Reg 00h[Bit5:4]<br>00/11: GPIO port<br>01: AC97 port<br>10: HD Audio port | Extend Reg 04h[Bit 2]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 02h[Bit 2] | Extend Reg 06h[Bit 2] | S0 |
| GPIO45/ AC_RST# | PM IO Reg59h[Bit 1:0]<br>00/01/10: AC97 port<br>11: GPIO | Extend Reg 05h[Bit 0]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 03h[Bit 0] | Extend Reg 07h[Bit 0] | S0 |
| GPIO46/ AZ_SDIN3 | SMBus Extend Reg 00h[Bit7:6]<br>00/11: GPIO port<br>01: N/A<br>10: HD Audio port | Extend Reg 04h[Bit 3]<br>0: Output enable<br>1: Input (Tri-state) | Extend Reg 02h[Bit 3] | Extend Reg 06h[Bit 3] | S0 |
| GPIO47/ SPI_CLK | SMBus RegA7h[Bit 6]<br>0: SPI_CLK<br>1: GPIO | Reg A6h[Bit 6]<br>0: Output<br>1: Input (Tri-state) | Reg A7h[Bit 2] | Reg A6h[Bit 2] | S0 |
| GPIO48/ FANOUT1 | PM IO Reg60h[Bit 2]<br>0: GPIO<br>1: FANOUT1 | Reg A6h[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg A7h[Bit 3] | Reg A6h[Bit 3] | S0 |
| GPIO49/ FANOUT2 | PM IO Reg60h[Bit 3]<br>0: GPIO<br>1: FANOUT2 | Reg 50h[Bit 4]<br>0: Output<br>1: Input (Tri-state) | Reg 51h[Bit 0] | Reg 50h[Bit 0] | S0 |
| GPIO50/ FANIN0 | PM2 IO Reg31h[Bit 0]<br>0: GPIO<br>1: FANIN0 | Reg 50h[Bit 5]<br>0: Output<br>1: Input (Tri-state) | Reg 51h[Bit 1] | Reg 50h[Bit 1] | S0 |
| GPIO51/ FANIN1 | PM2 IO Reg36h[Bit 0]<br>0: GPIO<br>1: FANIN1 | Reg 50h[Bit 6]<br>0: Output<br>1: Input (Tri-state) | Reg 51h[Bit 2] | Reg 50h[Bit 2] | S0 |
| GPIO52/ FANIN2 | PM2 IO Reg3Bh[Bit 0]<br>0: GPIO<br>1: FANIN2 | Reg 50h[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg 51h[Bit 3] | Reg 50h[Bit 3] | S0 |

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| GPIO[60:53]/ VIN[7:0] | PM2 IO Reg57h[Bit 7:6] [Bit5:4] Bit[3:2] Bit 1:0] Reg56h[Bit 7:6] [Bit5:4] Bit[3:2] Bit 1:0]<br>00: GPIO<br>01/10/11: VIN | Reg 54h[Bit 7:4]<br>Reg 52h[Bit 7:4]<br>0: Output<br>1: Input (Tri-state) | Reg 55h[Bit 3:0]<br>Reg 53h[Bit 3:0] | Reg 54h[Bit 3:0]<br>Reg 52h[Bit 3:0] | S0 |
| GPIO[63:61]/ TEMPIN[2:0] | PM2 IO Reg42h[Bit 5:4] [Bit 3:2] [Bit 1:0]<br>00: GPIO<br>01/10/11: TEMPIN | Reg 56h[Bit 6:4]<br>0: Output<br>1: Input (Tri-state) | Reg 57h[Bit 2:0] | Reg 56h[Bit 2:0] | S0 |
| GPIO64/ TEMPIN3/ TALERT# | PM2 IO Reg42h[Bit 7:6]<br>00: GPIO or TALERT#<br>01/10/11: TEMPIN3 | Reg 56h[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg 57h[Bit 3] | Reg 56h[Bit 3] | S0 |
| GPIO65/ BMREQ#/ REQ5# | PCIB Reg64h[Bit 8]<br>0: GPIO or BMREQ#<br>1: REQ5#<br>SMBus Reg64h[Bit 5]<br>0: GPIO<br>1: BMREQ# | Reg 7Eh[Bit 4]<br>0: Output<br>1: Input (Tri-state) | Reg 7Fh[Bit 0] | Reg 7Eh[Bit 0] | S0 |
| GPIO66/ LLB# | PM IO Reg68h[Bit 5]<br>0: GPIO<br>1: LLB# | Reg 7Eh[Bit 5]<br>0: Output<br>1: Input (Tri-state) | Reg 7Fh[Bit 1] | Reg 7Eh[Bit 1] | S5 |
| GPIO67/ SATA_ACT# | SMBus RegACh[Bit 3]<br>0: SATA_ACT#<br>1: GPIO | Reg ACh[Bit 1]<br>0: Output<br>1: Input (Tri-State) | Reg ACh[Bit 2] | Reg ACh[Bit 0] | S0 |
| GPIO68/ LDRQ1#/ GNT5# | PCIB Reg64h[Bit 8]<br>0: GPIO if not used by LPC<br>1: GNT5# | Reg 7Eh[Bit 6]<br>0: Output<br>1: Input (Tri-State) | Reg 7Fh[Bit 2] | Reg 7Eh[Bit 2] | S0 |
| GPIO69/ RTC_IRQ# | GPIO if internal RTC | Reg 7Eh[Bit 7]<br>0: Output<br>1: Input (Tri-State) | Reg 7Fh[Bit 3] | Reg 7Eh[Bit 3] | S5 |
| GPIO70/ REQ3# | SMBus Reg5Bh[Bit 4]<br>0: REQ3#<br>1: GPIO | Reg 5Ah[Bit 4]<br>0: Output<br>1: Input (Tri-state) | Reg 5Bh[Bit 0] | Reg 5Ah[Bit 0] | S0 |
| GPIO71/ REQ4# | SMBus Reg5Bh[Bit 5]<br>0: REQ4#<br>1: GPIO | Reg 5Ah[Bit 5]<br>0: Output<br>1: Input (Tri-state) | Reg 5Bh[Bit 1] | Reg 5Ah[Bit 1] | S0 |
| GPIO72/ GNT#3 | SMBus Reg5Bh[Bit 6]<br>0: GNT3#<br>1: GPIO | Reg 5Ah[Bit 6]<br>0: Output<br>1: Input (Tri-state) | Reg 5Bh[Bit 2] | Reg 5Ah[Bit 2] | S0 |
| GPIO73/ GNT4# | SMBus Reg5Bh[Bit 7]<br>0: GNT4#<br>1: GPIO | Reg 5Ah[Bit 7]<br>0: Output<br>1: Input (Tri-state) | Reg 5Bh[Bit 3] | Reg 5Ah[Bit 3] | S0 |

| Pin Name (Note 1) | Multi-function Selection | Output Enable (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Input if GPI (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Output if GPO (On SMBus Controller) Bus 00h/ Dev14h/ Fun00 | Power Domain |
|---|---|---|---|---|---|
| **Notes:**<br>**1–**In this table, the "GPIO" portion of the pin name has been put at the front of the names for the sake of clarity, making the pin names different from how they appear in the *AMD SB600 Databook*.<br>**2–**Register A9h[7:0] is addressed as A8[15:8] in some AMD documents.<br>**3–**Register AAh[7:0] is addressed as A8[23:16] in some AMD documents.<br>**4–**Register BDh[7:0] is addressed as BCh[15:8] in some AMD documents.<br>**5–**Register 0A2h[15:0] is addressed as 0A0h[31:16] in some AMD documents. | | | | | |

### *4.1.2 GPOC*

The two pairs of GPOC[3:2] and GPOC[1:0] pins are multi-purpose pins. They can be used as SMBus data and clock pins or general purpose input/output pins. When used as output pins, they are open collectors and need pull-up resistors for output high. Input/output programming is accomplished through the register pair C50h/C51h in index/data mode.

GPOC[3:2] pins are in the S5 power plane. GPOC[1:0] pins are in the S0 power plane.

The index register 12h is used to access GPOC pins and is defined as follows:

**Table 4-2: GPOC Pins**

| IO C50h/C51h index 12h Bit | Field Name | Default | Description |
|---|---|---|---|
| 0 | GPOC0 Status | -- | GPOC0 Input read status |
| 1 | GPOC1 Status | -- | GPOC1 Input read status |
| 2 | GPOC0_OE | 1 | 0 = GPOC0 is asserted low<br>1 = GPOC0 is tri-state |
| 3 | GPOC1_OE | 1 | 0 = GPOC1 is asserted low<br>1 = GPOC1 is tri-state |
| 4 | GPOC2 Status | -- | GPOC2 Input read status |
| 5 | GPOC3 Status | -- | GPOC3 Input read status |
| 6 | GPOC2_OE | 1 | 0: GPOC2 is asserted low<br>1: GPOC2 is tri-state |
| 7 | GPOC3_OE | 1 | 0: GPOC3 is asserted low<br>1: GPOC3 is tri-state |

## 4.2 GEVENT/GPE/GPM/ExtEvent

### *4.2.1 GEVENT as GPIO*

GEVENT[1:0] are inputs only. Their status can be read from PM I/O Reg 92h Bit[1:0].

GEVENT[7:2] can be programmed either as GPIO lines or as GPE lines (see *section 4.2.2*).

- PM I/O Reg 91h Bit[7:2] defines GEVENT[7:2] pins as output or input
  - 0: Enable output
  - 1: Tris-tate (input)
- For GEVENT pins defined as output, PM I/O Reg 90h Bit[7:2] sets the output value
  - 0: Output low
  - 1: Output high
- For GEVENT pins defined as input, PM I/O Reg 92h Bit[7:2] are used to read the input status.

GEVEN8 can be programmed either as GPIO line or as GPE line (see *section 4.2.2*).

- PM I/O Reg 84h Bit[3] defines GEVENT8 pins as output or input
  - 0: Enable output
  - 1: Tris-tate (input)
- For GEVENT8 pin defined as output, PM I/O Reg 84h Bit[2] sets the output value
  - 0: Output low
  - 1: Output high
- For GEVENT8 pin defined as input, PM I/O Reg 84h Bit[4] is used to read the input status.

### *4.2.2 General Purpose Event (GPE)*

The General Purpose Event (GPE) pins can generate wake events, SCI, SMI#, SMI# followed by SCI, or IRQ13.

When not used as GPE pins, these pins may also be used as GPIO pins as described in *sections 4.1.1*, *4.2.1*, *4.2.3*, *4.2.4*.

The GPE pins include:

- GEVENT[8:0]
- GPM[9:0]
- EXTEVENT[1:0]
- GPIO0, GPIO2, GPIO64

**Table 4-3: GPE Pins**

| Pin Name (*Note 1) | Multi-Function Selection | Configure Bit<br>00 – SCI or SMI#<br>01 – SMI#<br>10 – SMI# followed by SCI<br>11 - IRQ13 | Trigger Configure<br>0–Falling edge<br>1–Rising edge | Enable ACPI Event | Status (Write 1 to ACPI GPE00h Bit to Clear) | Power Domain |
|---|---|---|---|---|---|---|
| GEVENT0/ GA20IN | SMBus Reg64h[Bit11]<br>0: GEVENT0<br>1: GA20IN | PM IO Reg30h[Bit1:0] | PM IO Reg36h[Bit 0] | ACPI GPE04h[Bit 0] | PM IO Reg39h[Bit 0]<br>or ACPI GPE00h[Bit0] | S0 |
| GEVENT1/ KBRST# | SMBus Reg64h[Bit9]<br>0: KBRST#<br>1: GEVENT1 | PM IO Reg30h[Bit3:2] | PM IO Reg36h[Bit 1] | ACPI GPE04h[Bit 1] | PM IO Reg39h[Bit 1]<br>or ACPI GPE00h[Bit 1] | S0 |

| Pin Name (*Note 1) | Multi-Function Selection | Configure Bit<br>00 – SCI or SMI#<br>01 – SMI#<br>10 – SMI# followed by SCI<br>11 - IRQ13 | Trigger Configure<br>0–Falling edge<br>1–Rising edge | Enable ACPI Event | Status (Write 1 to ACPI GPE00h Bit to Clear) | Power Domain |
|---|---|---|---|---|---|---|
| GEVENT2/ THRMTRIP#/SMBALERT# | PM IO Reg68h[Bit 3]<br>0: GEVENT2<br>1: THRMTRIP# | PM IO Reg30h[Bit5:4] | PM IO Reg36h[Bit 2] | ACPI GPE04h[Bit 2] | PM IO Reg39h[Bit 2]<br>or ACPI GPE00h[Bit 2] | S5 |
| GEVENT3/ LPC_PME# | ***Note 5** | PM IO Reg30h[Bit7:6] | PM IO Reg36h[Bit3] | ACPI GPE04h[Bit3] | PM IO Reg39h[Bit3]<br>or ACPI GPE00h[Bit3] | S5 |
| GEVENT4/ PCI_PME# | ***Note 5** | PM IO Reg31h[Bit1:0] | PM IO Reg36h[Bit4] | ACPI GPE04h[Bit4] | PM IO Reg39h[Bit4]<br>or ACPI GPE00h[Bit4] | S5 |
| GEVENT5/ S3_STATE | SMBus Reg64h[Bit 8]<br>=1 to enable GPE<br>***Note 4** | PM IO Reg31h[Bit3:2] | PM IO Reg36h[Bit 5] | ACPI GPE04h[Bit 5] | PM IO Reg39h[Bit 5]<br>or ACPI GPE00h[Bit 5] | S5 |
| GEVENT6/ USB_OC6# | ***Note 5** | PM IO Reg31h[Bit5:4] | PM IO Reg36h[Bit 6] | ACPI GPE04h[Bit 6] | PM IO Reg39h[Bit 6]<br>or ACPI GPE00h[Bit 6] | S5 |
| GEVENT7/ USB_OC7# | ***Note 5** | PM IO Reg31h[Bit7:6] | PM IO Reg36h[Bit 7] | ACPI GPE04h[Bit 7] | PM IO Reg39h[Bit 7]<br>or ACPI GPE00h[Bit 7] | S5 |
| GEVENT8/ WAKE# | PM IO Reg84h[Bit 0]<br>0: GEVENT8<br>1: WAKE#<br>PM IO Reg84h[Bit 1]=1 and SMBus Reg 64h[Bit 19]=1 to enable GPE | PM IO Reg34h[Bit1:0] | PM IO Reg38h[Bit 0] | ACPI GPE04h[Bit24] | PM IO Reg3Bh[Bit 0]<br>or ACPI GPE00h[Bit24] | S5 |
| GPM0/ USB_OC0# | SMBus Reg64h[Bit 23]<br>=1 to enable GPE | PM IO Reg32h[Bit7:6] | PM IO Reg37h[Bit 3] | ACPI GPE04h[Bit19] | PM IO Reg3Ah[Bit 3]<br>or ACPI GPE00h[Bit19] | S5 |
| GPM1/ USB_OC1# | SMBus Reg64h[Bit 23]<br>=1 to enable GPE | PM IO Reg33h[Bit1:0] | PM IO Reg37h[Bit 4] | ACPI GPE04h[Bit20] | PM IO Reg3Ah[Bit 4]<br>or ACPI GPE00h[Bit20] | S5 |
| GPM2/ USB_OC2# | SMBus Reg64h[Bit 23]<br>=1 to enable GPE | PM IO Reg33h[Bit3:2] | PM IO Reg37h[Bit 5] | ACPI GPE04h[Bit21] | PM IO Reg3Ah[Bit 5]<br>or ACPI GPE00h[Bit21] | S5 |

| Pin Name (*Note 1) | Multi-Function Selection | Configure Bit<br>00 – SCI or SMI#<br>01 – SMI#<br>10 – SMI# followed by SCI<br>11 - IRQ13 | Trigger Configure<br>0–Falling edge<br>1–Rising edge | Enable ACPI Event | Status (Write 1 to ACPI GPE00h Bit to Clear) | Power Domain |
|---|---|---|---|---|---|---|
| GPM3/<br>USB_OC3# | SMBus Reg64h[Bit 23] =1 to Enable GPE | PM IO Reg33h[Bit5:4] | PM IO Reg37h[Bit 6] | ACPI GPE04h[Bit22] | PM IO Reg3Ah[Bit 6]<br>or ACPI GPE00 [Bit 22] | S5 |
| GPM4/<br>USB_OC4# | SMBus Reg64h[Bit 19] =1 to enable GPE | PM IO Reg34h[Bit3:2] | PM IO Reg38h[Bit 1] | ACPI GPE04h[Bit25] | PM IO Reg3Bh[Bit 1]<br>or ACPI GPE00h[Bit25] | S5 |
| GPM5/<br>USB_OC5# | SMBus Reg64h[Bit 19]<br>=1 to enable GPE | PM IO Reg34h[Bit5:4] | PM IO Reg38h[Bit 2] | ACPI GPE04h[Bit26] | PM IO Reg3Bh[Bit 2]<br>or ACPI GPE00h[Bit26] | S5 |
| GPM6/<br>BLINK | PM IO Reg7Ch[Bit 3:2]<br>00: GPM6<br>01/10/11: BLINK 1/4Hz, 1/2Hz, and always-on<br>SMBus Reg64h[Bit 18]<br>=1 to enable GPE | PM IO Reg35h[Bit1:0] | PM IO Reg38h[Bit 4] | ACPI GPE04h[Bit28] | PM IO Reg3Bh[Bit 4]<br>or ACPI GPE00h[Bit28] | S5 |
| GPM7/<br>SYS_<br>RESET# | PM IO Reg55h[Bit 2]<br>0: GPM7<br>1: SYS_RESET#<br>SMBus Reg64h[Bit 17]<br>=1 to enable GPE | PM IO Reg35h[Bit3:2] | PM IO Reg38h[Bit 5] | ACPI GPE04h[Bit29] | PM IO Reg3Bh[Bit 5]<br>or ACPI GPE00h[Bit29] | S5 |
| GPM8/<br>USB_OC8#/<br>AZ_DOCK_<br>RST# | PM IO Reg8Dh[Bit 2]<br>0: GPM8<br>1: AZ_DOCK_RST#<br>SMBus Reg64h[Bit 23]<br>=1 to enable GPE | PM IO Reg33h[Bit7:6] | PM IO Reg37h[Bit 7] | ACPI GPE04h[Bit23] | PM IO Reg3Ah[Bit 7]<br>or ACPI GPE00 [Bit 23] | S5 |
| GPM9/<br>USB_OC9#/<br>SLP_S2 | PM IO Reg8Dh[Bit 1]<br>0: GPM9<br>1: SLP_S2 | PM IO Reg3Dh[Bit3:2] | Always falling edge | ACPI GPE04h[Bit14] | ACPI GPE00 [Bit 14] | S5 |
| EXTEVENT<br>0#/<br>RI# | SMBus Reg64h[Bit 22]<br>=1 to enable GPE | PM IO Reg32h[Bit1:0] | PM IO Reg37h[Bit 0] | ACPI GPE04h[Bit16] | PM IO Reg3Ah[Bit 0]<br>or ACPI GPE00h[Bit16] | S5 |

| Pin Name (*Note 1) | Multi-Function Selection | Configure Bit<br>00 – SCI or SMI#<br>01 – SMI#<br>10 – SMI# followed by SCI<br>11 - IRQ13 | Trigger Configure<br>0–Falling edge<br>1–Rising edge | Enable ACPI Event | Status (Write 1 to ACPI GPE00h Bit to Clear) | Power Domain |
|---|---|---|---|---|---|---|
| EXTEVENT 1#/ LPC_SMI# | SMBus Reg64h[Bit 22] =1 to enable GPE | PM IO Reg32h[Bit3:2] | PM IO Reg37h[Bit 1] | ACPI GPE04h[Bit17] | PM IO Reg3Ah[Bit 1] or ACPI GPE00h[Bit17] | S0 |
| GPIO0 | PM IO Reg60h[Bit 7]=1 for GPIO; in addition, PM IO Reg84h[Bit 1]=0 and SMBus Reg64h[Bit 19]=1 to enable GPE | PM IO Reg34h[Bit1:0] | PM IO Reg38h[Bit 0] | ACPI GPE04h[Bit24] | PM IO Reg3Bh[Bit 0] or ACPI GPE00h[Bit24] | S0 |
| GPIO2 | PM IO Reg60h[Bit 5]=0 for GPIO | PM IO Reg35h[Bit5:4] | PM IO Reg38h[Bit 6] | ACPI GPE04h[Bit 30] | PM IO Reg3Bh[Bit 6] or ACPI GPE00h[Bit 30] | S0 |
| GPIO64/ TALERT#/ TEMPIN3 | PM2 IO Reg42h[Bit 7:6]<br>00: GPIO<br>01/10/11: TEMPIN3 | PM IO Reg3Ch[Bit3:2]<br>SMI# followed by SCI not available | PM IO Reg67h[Bit 5] | ACPI GPE04h[Bit 9] | ACPI GPE00h[Bit 9] | S0 |

**Notes:**

**1–** In this table, the "GEVENT," "GPM," "EXTEVNT," or "GPIO" portion of the pin name has been put at the front of the names for the sake of clarity, making the pin names different from how they appear in the *AMD SB600 Databook*.

**2–**PM IO Register can be accessed through IO port CD6h/CD7h.

**3–**GPE Register is in the ACPI IO space. The base address of GPE IO space is defined in PM IO Reg28h/29h.

**4–**In K8 system, this pin is always used as S3_STATE output (indicating ACPI S3 state).

**5–**This pin is GEVENT pin only and not a multiplexed pin. The alternative function is mentioned only as a suggestion.

### *4.2.3 GPM as GPIO*

GPM pins can be used as GPIO. The GPM I/O function is controlled by three registers: I/O C50h, C51h, C52h, PM I/O 94h, 95h, 96h.

#### *4.2.3.1 GPM Pins as Input*

For GPM[7:0], follow this sequence -

1. Set index register 0C50h to 13h (Miscellaneous Control).
2. Set CM Data register 0C51h Bits [7:6] to 01b to set Input/Out control.
3. Set GPM port 0C52h appropriate bits to 1 to tri-state the GPM port.
4. Set CM Data register 0C51h Bits [7:6] = 00b to set GPM port for read.
5. Read the input status through port 0C52h.

For GPM[9:8], simply read the input status from PM I/O 96h Bits [1:0].

#### 4.2.3.2 GPM pins as Output

For GPM[7:0], follow this sequence -

1. Set index register 0C50h to 13h (Misc. Control).
2. Set CM Data register 0C51h Bits [7:6] = 01b to set Input/Out control.
3. Set GPM port 0C52h appropriate bits to 0 to enable output on the GPM port.
4. Set CM Data register 0C51h Bits [7:6] = 10b for output data control.
5. Write the output data to port 0C52h.

For GPM[9:8], simply use PM I/O 95h Bits [1:0] as output enable (0: enable; 1: tri-state) and 94h Bits [1:0] as output value (0: output low; 1: output high).

### 4.2.4 ExtEvent

#### 4.2.4.1 ExtEvent as GPIO

**Table 4-4: ExtEvent Pins as GPIO**

| Pin Name | Output Enable | Output Register | Input Status |
|---|---|---|---|
| EXTEVNT0#/ RI# | PM IO Reg91h[Bit 0]<br>0: Output enable<br>1: Input (Tri-state) | PM IO Reg90h[Bit 0]<br>0: Output low<br>1: Output high | PM IO Reg0Dh[Bit 0] |
| EXTEVNT1#/ LPC_SMI# | PM IO Reg91h[Bit 1]<br>0: Output enable<br>1: Input(Tri-state) | PM IO Reg90h[Bit 1]<br>0: Output low<br>1: Output high | PM IO Reg0Dh[Bit 1] |

#### 4.2.4.2 ExtEvent to Generate SMI#

EXTEVNT[1:0] can generate SCI or SMI#/SMI#/SMI# followed by SCI/IRQ13 as described in section 4.2.2. These pins can also be programmed to generate SMI# by the following alternate method.

**Table 4-5: ExtEvent Pins to Generate SMI#**

| Pin Name* | Trigger<br>1 – Rising edge<br>0 – Falling edge | Enable SMI# | SMI# Status | Logical Value Read only | Power Domain |
|---|---|---|---|---|---|
| EXTEVNT0#/ RI# | PM IO Reg0Dh[Bit 2] | PM IO Reg04h[Bit 0] | PM IO Reg07h[Bit 0] | PM IO Reg0Dh[Bit 0] | S5 |
| EXTEVNT1#/ LPC_SMI# | PM IO Reg0Dh[Bit 3] | PM IO Reg04h[Bit 1] | PM IO Reg07h[Bit 1] | PM IO Reg0Dh[Bit 1] | S0 |

# 4.3 THRMTRIP/TALERT

### *4.3.1 Thermal Trip – THRMTRIP*

The thermal trip function is multiplexed on the GEVENT2 pin. The THRMTRIP status cannot be used to generate SCI or SMI#.

**Table 4-6: THRMTRIP Pin**

| Pin Name | Enable THRMTRIP | THRMTRIP Polarity Control | THRMTRIP Status |
|---|---|---|---|
| THRMTRIP#/ GEVENT2#/ SMBALERT# | PM IO Reg68h[Bit 3]=1 in addition PM IO Reg55h[Bit 0]=1 to enable shutdown | PM IO Reg67h[Bit 6] 0=Active Low 1=Active High | PM IO Reg54h[Bit 0] |

### *4.3.2 Temperature Alert – TALERT*

The temperature alert function is multiplexed on the GPIO64 pin. It can be programmed to generate SMI#, SCI, or IRQ13 through GPE. It can also be programmed to generate SMI# without using GPE.

**Table 4-7: TALERT# through GPE**

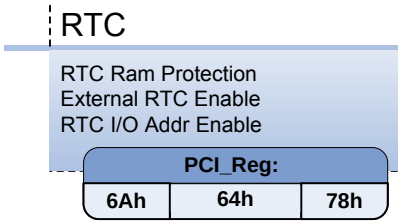
| Pin Name | TALERT# Enable | TAERLT# Polarity Control | TALERT# Status | TALERT# Outcome |
|---|---|---|---|---|
| TALERT#/ GPIO64/ TEMPIN3 | AcpiGpe0Blk04[Bit 9]=1 | PM IO Reg67h[Bit 5] 0=Active Low 1=Active High | AcpiGpe0Blk00[Bit 9] | Controlled by PM I/O Reg3Ch[Bit3:2]: 00= ACPI Event (SCI or SMI#, depending on SCI_EN bit) 01= SMI# 10= N/A 11= IRQ13 |

**Table 4-8: TALERT# to generate SMI#**

| Pin Name | TALERT# SMI# Enable | TAERLT# Polarity Control | TALERT# Status | TALERT# Outcome |
|---|---|---|---|---|
| TALERT#/ GPIO64/ TEMPIN3 | IO C50h/C51h, index 03h [Bit 1]=1 | PM IO Reg67h[Bit 5] 0=Active Low 1=Active High | IO C50h/C51h, index 02h [Bit 0] | SMI# |

## 4.4 Real Time Clock (RTC)

Note: Some RTC functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these RTC functions and the associated registers.




For software compatibility, the RTC registers and RAM are accessed through two banks addressing – Bank 0 and Bank 1, which are shown in the diagram below. Bank 0 is selected if DV0 = 0(DV0 is the 5$^{th}$ bit of Register A) while Bank 1 is chosen if DV0 = 1. Even if two banks are defined, the first 64 bytes (00H – 3FH) of each bank share the same physical space. The SB600 has alternate RAM ports at h72/h73 as the address/data ports. **These two ports do not use the bank0/bank1 scheme. Memory can be accessed directly using the 8-bit address port. Use of the two IO ports h72/h73 to access the RTC RAM is highly recommended.**

**Note:** Some RTC RAM space can be protected from read/write if corresponding bits are set to 1 in RTCProtect register (PCI_Reg 6Ah).

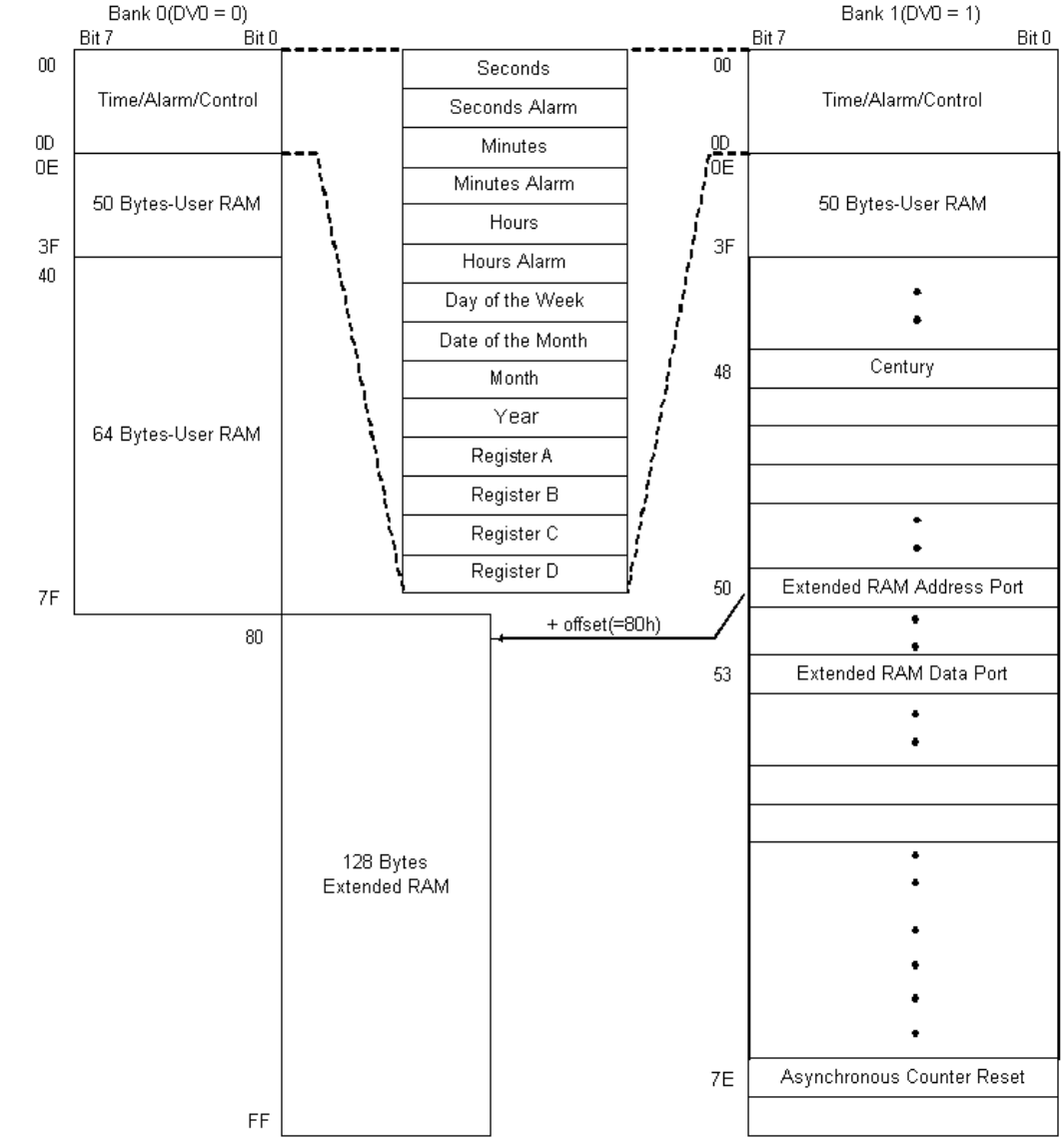



**Figure 5 Register Bank Definition and Memory Address Mapping**

The analog portion consists of two major parts: one is a 256-byte CMOS RAM and the other a 44-bit ripple counter.

| Register Name | Offset Address |
|---|---|
| Seconds | 00h |
| Seconds Alarm | 01h |
| Minutes | 02h |
| Minutes Alarm | 03h |
| Hours | 04h |
| Hours Alarm | 05h |
| Day of Week | 06h |
| Date of Month | 07h |
| Month | 08h |
| Year | 09h |

| Register Name | Offset Address |
|---|---|
| Register A | 0Ah |
| Register B | 0Bh |
| Register C | 0Ch |
| Register D | 0Dh |
| AltCentury (when DV0=0) | 32h |
| Century (when DV0=1) | 48h |
| Extended RAM Address Port | 50h |
| Extended RAM Data Port | 53h |
| RTC Time Clear | 7Eh |
| RTC RAM Enable | 7Fh |

Registers that are implemented in the internal RTC are described below.

| Seconds - RW – 8 bits - [RTC_Reg: 00h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Seconds | 7:0 | 00h | Binary-Code-Decimal format. Range:00 – 59<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every second. When set by software, hardware updating is disabled. |
| Seconds register | | | |

| Seconds Alarm - RW – 8 bits - [RTC_Reg: 01h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Seconds Alarm | 7:0 | 00h | Binary-Code-Decimal format.<br>If SET bit = 1, Seconds Alarm Register will never match with Seconds Register, else<br>If bits [7:6] = [11], Seconds Alarm Register always matches with Seconds Register. |
| Seconds Alarm register | | | |

| Minutes - RW – 8 bits - [RTC_Reg: 02h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minutes | 7:0 | 00h | Binary-Code-Decimal format. Range:00 – 59<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every minute. When set by software, hardware updating is disabled. |
| Minutes register | | | |

| Minutes Alarm - RW – 8 bits - [RTC_Reg: 03h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Minutes Alarm | 7:0 | 00h | Binary-Code-Decimal format.<br>If SET bit = 1, Minutes Alarm Register will never match with Minutes Register, else<br>If bits [7:6] = [11], Minutes Alarm Register always matches with Minutes Register. |
| Minutes Alarm register | | | |

| Hours - RW – 8 bits - [RTC_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Hours | 7:0 | 00h | Binary-Code-Decimal format. Range:00 – 23<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every hour. When set by software, hardware updating is disabled. |

| Hours - RW – 8 bits - [RTC_Reg: 04h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Hours register | | | |

| Hours Alarm- RW – 8 bits - [RTC_Reg: 05h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Hours Alarm | 7:0 | 00h | Binary-Code-Decimal format.<br>If SET bit = 1, Hours Alarm Register will never match with Hours Register, else<br>If bits [7:6] = [11], Hours Alarm Register always matches with Hours Register. |
| Hours Alarm register | | | |

| Day of Week - RW – 8 bits - [RTC_Reg: 06h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Day of Week | 7:0 | 00h | Binary-Code-Decimal format. Range: 01 – 07 (Sunday = 1).<br>No leap year correction capability. Leap year correction has to be done by software.<br>This register can be set by a software (SET bit of Register B = 1) or can be automatically updated by hardware everyday.<br>When set by software, hardware updating is disabled. |
| Day of Week register | | | |

| Date of Month - RW – 8 bits - [RTC_Reg: 07h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Date of Month | 7:0 | 00h | Binary-Code-Decimal format. Range: 01 – 28 for February and no leap year capability. Leap year correction has to be done by software.<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware everyday. When set by software, hardware updating is disabled. |
| Date of Month register | | | |

| Month - RW – 8 bits - [RTC_Reg: 08h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Month | 7:0 | 00h | Binary-Code-Decimal format. Range: 01 – 12. No leap year correction capability. Leap year correction has to be done by software.<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every month.<br>When set by software, hardware updating is disabled. |
| Month register | | | |

| Year - RW – 8 bits - [RTC_Reg: 09h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Year | 7:0 | 00h | Binary-Code-Decimal format. Range: 00 – 99. No leap year correction capability. Leap year correction has to be done by software.<br>This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every year.<br>When set by software, hardware updating is disabled. |
| Year register | | | |

| Register A - RW – 8 bits - [RTC_Reg: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rate Selection(RS0) | 0 | 0b | These four rate-selection bits select one of the 13 taps on the |

| Register A - RW – 8 bits - [RTC_Reg: 0Ah] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Rate Selection(RS1) | 1 | 0b | 15-stage frequency divider or disable the divider output (flat output signal). The tap selected can be used to generate a periodic interrupt. See the following table for frequency selection. |
| Rate Selection(RS2) | 2 | 0b | |
| Rate Selection(RS3) | 3 | 0b | |
| Bank Selection(DV0) | 4 | 0b | DV0 = 0 selects Bank 0; DV0 = 1 selects Bank 1. The SB600, it has an alternate way to access the RAM without the use of bank select bit. Port 72/73 can be used as the index to access the full 256 bytes of RAM directly. |
| Reserved | 6:5 | | |
| Update In Progress(UIP) | 7 | 0b | If SET bit = 1, UIP is cleared.<br>If UIP = 1, the update transfer will soon occur.<br>If UIP = 0, the update transfer will not occur for at least 244us.<br>[Read-only] |
| Register A: Control register | | | |

| Rate Selection Bits | | | | Tap Frequency(Interrupt Rate) |
|---|---|---|---|---|
| **RS3** | **RS2** | **RS1** | **RS0** | |
| 0 | 0 | 0 | 0 | Flat Signal(None) |
| 0 | 0 | 0 | 1 | 256 Hz (3.90625 ms) |
| 0 | 0 | 1 | 0 | 128 Hz (7.8125 ms) |
| 0 | 0 | 1 | 1 | 8.192 kHz (122.070 us) |
| 0 | 1 | 0 | 0 | 4.096 kHz (244.141 us) |
| 0 | 1 | 0 | 1 | 2.048 kHz (488.281 us) |
| 0 | 1 | 1 | 0 | 1.024 kHz (976.5625 us) |
| 0 | 1 | 1 | 1 | 512 Hz (1.953125 ms) |
| 1 | 0 | 0 | 0 | 256 Hz (3.90625 ms) |
| 1 | 0 | 0 | 1 | 128 Hz (7.8125 ms) |
| 1 | 0 | 1 | 0 | 64 Hz (15.625 ms) |
| 1 | 0 | 1 | 1 | 32 Hz (31.25 ms) |
| 1 | 1 | 0 | 0 | 16 Hz (62.5 ms) |
| 1 | 1 | 0 | 1 | 8 Hz (125 ms) |
| 1 | 1 | 1 | 0 | 4 Hz (250 ms) |
| 1 | 1 | 1 | 1 | 2 Hz (500 ms) |

| Register B - RW – 8 bits - [RTC_Reg: 0Bh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 0 | 0b | |
| HourMode | 1 | 0b | Hour mode; 0 = 12 hour mode; 1 = 24 hour mode |
| Reserved | 3:2 | 00 | |
| Update Ended Interrupt Enable(UIE) | 4 | 0b | UIE enables the Update End Flag (UF) bit in Register C to assert IRQ. If SET bit = 1, UIE is cleared. |
| Alarm Interrupt Enable (AIE) | 5 | 0b | AIE enables the Alarm Flag (AF) bit in Register C to assert IRQ. |
| Periodic Interrupt Enable (PIE) | 6 | 0b | PIE enables the Periodic Interrupt Flag (PF) bit in Register C to assert IRQ. |
| Set new time (SET) | 7 | 0b | If SET bit = 1, no internal updating for Time Registers is allowed.<br>If SET bit = 0, the Time Registers are updated every second. |
| Register B: Control register | | | |

| Register C - R – 8 bits - [RTC_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Reserved | 3:0 | 0h | |

| Register C - R – 8 bits - [RTC_Reg: 0Ch] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Update Ended Interrupt Flag(UF) | 4 | 0b | This bit is set to one after each update cycle. Reading Register C clears UF. |
| Alarm Interrupt Flag (AF) | 5 | 0b | This bit is set to one if second, minute and hour time has matched the second, minute and hour alarm time. Reading Register C clears AF bit. |
| Periodic Interrupt Flag (PF) | 6 | 0b | This bit is set to one when an edge is detected on the selected tap (through RS3 to RS0) of the frequency divider. Reading Register C clears PF bit. |
| Interrupt Request Flag (IRQF) | 7 | 0b | Logically, IRQF = (PF*PIE)+(AF*AIE)+(UF*UIE)+(WF*WIE) where WF and WIE are defined in Extended Control Register 4A and 4B. Reading Register C clears IRQF bit. Any time the IRQF bit is set to one, the #IRQ pin is driven low. |
| Register C: Control register | | | |

| DateAlarm - RW – 8 bits - [RTC_Reg: 0Dh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| DateAlarm | 5:0 | 00h | DateAlarm in BCD format and is considered when it is set to non-zero value. If this value is set to 0, then date is not compared for alarm generation. |
| Scratchbit | 6 | 0b | |
| VRT | 7 | 1b | Valid RAM and Time; refer to VRT_T1 and VRT_T2 registers (PMIO 3E/3F) |
| Date Alarm Register | | | |

| AltCentury - RW – 8 bits - [RTC_Reg: 32h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| AltCentury | 7:0 | 00h | (This register is accessed only when DV0=0 and PM_Reg 7Ch Bit4=1.) Binary-Code-Decimal format. Leap year correction is done through hardware. This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every century. When set by software, hardware updating is disabled. |
| AltCentury Register | | | |

| Century - RW – 8 bits - [RTC_Reg: 48h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Century | 7:0 | 00h | (This register is accessed only when DV0=1) Binary-Code-Decimal format. Leap year correction is done through hardware. This register can be set by software (SET bit of Register B = 1) or can be automatically updated by hardware every century. When set by software, hardware updating is disabled. |
| Century Register | | | |

| Extended RAM Address Port - RW – 8 bits - [RTC_Reg: 50h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| ExtendedRAMAddr | 6:0 | 00h | Because only 7 address bits are used in port x70, only lower 128 bytes are accessible through port x71. The Extended RAM (upper 128 bytes) are physically located at address 80H to FFH. In order to access these address, an address offset should be programmed into this register and access them through Extended RAMDataPort. (An offset of x80H will automatically add to this 7-bit address). |
| Reserved | 7 | | |
| Extended RAM Address Port register: The address port to access Extended RAM. | | | |

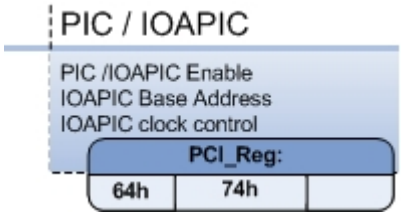
| Extended RAM Data Port - RW – 8 bits - [RTC_Reg: 53h] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Extended RAM Data Port | 7:0 | xxxxxxxx | There is no physical register corresponding to this data port but the data port address is used for decoding to generate appropriate internal control signals. |
| Extended RAM Data Port register. | | | |

| RTC Time Clear - RW – 8 bits - [RTC_Reg: 7Eh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RtcTimeClear | 0 | 0b | Setting this bit will clear the RTC second and RTC time will stop |
| Reserved | 7:1 | 0000000b | |
| RTC Time Clear register. | | | |

| RTC RAM Enable - RW – 8 bits - [RTC_Reg: 7Fh] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| RtcRamEnable | 0 | 1b | Setting this bit will enable access to the RTC RAM |
| Reserved | 7:1 | 0000000b | |
| RTC RAM Enable register. | | | |

## 4.5 IOXAPIC Registers

Note: Some IOXAPIC functions are controlled by, and associated with, certain PCI configuration registers in the SMBus/ACPI device. For more information refer to *section 2.3: SMBus Module and ACPI Block (Device 20, Function 0)*. The diagram below lists these IOXAPIC functions and the associated registers.




### *4.5.1 Direct Access Registers*

Note: The XAPIC_BASE_REGISTER has a power-on default value of FEC0_0000H.

| IO Register Select Register RW [XAPIC_BASE_REGISTER + 00H] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Indirect Address Offset | 7:0 | 00h | Indirect Address Offset to IO Window Register |
| Reserved | 31:8 | | |
| Used to determine which register is manipulated during an IO Window Register read/write operation. | | | |

| IO Window Register RW [XAPIC_BASE_REGISTER + 10H] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Mapped by the value in the IO Register Select Register, to the designated indirect access register. Technically a R/W register; however, the read/write capability is determined by the indirect access register referenced by the IO Register Select Register. | | | |

| IRQ Pin Assertion Register RW [XAPIC_BASE_REGISTER + 20H] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Input IRQ | 7:0 | 00h | IRQ number for the requested interrupt |
| <reserved> | 31:8 | 0000000h | |
| Write to this register will trigger an interrupt associated with the redirection table entry referenced by the IRQ number. Currently the redirection table has 24 entries. Write with IRQ number greater than 17H has no effect. | | | |

| EOI Register W [XAPIC_BASE_REGISTER + 40H] | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vector | 7:0 | 00h | Interrupt vector |
| <reserved> | 31:8 | 0000000h | |
| Write to this register will clear the remote IRR bit in the redirection table entry found matching the interrupt vector. This provides an alternate mechanism other than PCI special cycle for EOI to reach IOXAPIC. | | | |

### *4.5.2 Indirect Access Registers*

Software needs to first select the register to access using the IO Register Select Register, and then read or write using the IO Window Register.

| IOAPIC ID Register [Indirect Address Offset = 00H] RW | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| <reserved> | 23:0 | 000000h | |
| ID | 27:24 | 0h | IOAPIC device ID for APIC serial bus delivery mode |
| <reserved> | 31:28 | 0h | |
| Not used in XAPIC PCI bus delivery mode. | | | |

| IOXAPIC Version Register [Indirect Address Offset = 01H] R | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Version | 7:0 | 21h | PCI 2.2 compliant |
| <Reserved> | 14:8 | 00h | |
| PRQ | 15 | 1b | IRQ pin assertion supported |
| Max Redirection Entries | 23:16 | 17h | 24 entries [23:0] |
| <Reserved> | 31:24 | 00h | |

| IOAPIC Arbitration Register [Indirect Address Offset = 02H] R | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| <reserved> | 23:0 | 000000h | |
| Arbitration ID | 27:24 | 0h | Arbitration ID for APIC serial bus delivery mode |
| <reserved> | 31:28 | 0h | |
| Not used in XAPIC PCI bus delivery mode. | | | |

| Redirection Table Entry [0–23] [Indirect Address Offset = 11/10H–3F/3EH] RW | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Vector | 7:0 | 00h | Interrupt vector associated with this interrupt input |
| Delivery Mode | 10:8 | 0h | 000 – Fixed<br>001 – Lowest Priority<br>010 – SMI/PMI<br>011 – <reserved><br>100 – NMI<br>101 – INIT<br>110 – <Reserved><br>111 – ExtINT |
| Destination Mode | 11 | 0b | 0 – Physical<br>1 – Logical |
| Delivery Status | 12 | 0b | Read Only<br>0 – Idle<br>1 – Send Pending |
| Interrupt Pin Polarity | 13 | 0b | 0 – High<br>1 – Low |
| Remote IRR | 14 | 0b | Read Only.<br>Used for level triggered interrupts only.<br>Set when interrupt message delivered.<br>Cleared by EOI special cycle transaction or write to EOI register |
| Trigger Mode | 15 | 0b | 0 – Edge<br>1 – Level |
| Mask | 16 | 1b | Mask the interrupt injection at the input of this device<br>Write 0 to unmask |
| Reserved | 31:17 | 0000h | |
| Reserved | 55:32 | 000000h | |

| Redirection Table Entry [0–23] [Indirect Address Offset = 11/10H–3F/3EH] RW | | | |
|---|---|---|---|
| **Field Name** | **Bits** | **Default** | **Description** |
| Destination ID | 63:56 | 0 | Bits [19:12] of the address field of the interrupt message |

# Appendix A: AC97 Audio FAQs

Q: What is the descriptor table (DT) data structure in memory?
A: Data Pointer (first dword); Size, Status (2nd dword); Next descriptor pointer (3rd dword). The Data pointer points to the beginning of a data block. In the 2nd dword, Size uses [31:16], and Status uses bit[0]. The other bits are 0. Size tells the data block size pointed by the Data Pointer. The Status bit can be set to 1 by hardware when the current data block is completed. Software can check this bit to know the progress of the hardware and plan ahead. Another way to accomplish this is to set the status bit in the register so the software will check that register. The Next descriptor pointer points to the next descriptor table.

Q: Where is the status bit updated each time a data block is completed?
A: For audio input:
When reg0x04[1]=1 and reg0x08[3]=0: the status bit is updated in reg0x00[1]. When reg0x04[1]=1 and reg0x08[3]=1: the status bit is updated in memory.
For audio output:
When reg0x04[1]=1 and reg0x08[3]=0: the status bit is updated in reg0x00[3]. When reg0x04[1]=1 and reg0x08[3]=1: the status bit is updated in memory.

Q: I set the appropriate bits so that the interrupt register (0x00) has the “out DMA status” bit set at the end of a data block and an interrupt happens. Is this necessary in order to cause an interrupt to happen?
A: Yes.

Q: Is it possible to have both an interrupt occur and the status bit in memory set at the same time?
A: No. The Status bit can be set either in memory or in register, but not both.

Q: What is a Linked List?
A: When a series of DTs are linked, it becomes a Linked List whose last DT has an invalid value of 0 as its Next Descriptor to signal the end of the chain.

Q: How can I determine whether a Linked List is complete?
A: This is determined by examining bit 0 of the current DT control block status word in memory to see if the DMA controller is finished with it, and then you examine the “Output DMA DT Next” register (0x40) to see if this was the last one in the chain or not. When both conditions are satisfied, you can tell the Linked List is complete.

Q: I am trying to determine if the DMA controller is idle or not. It seems that the “Output DMA DT Size and State” register (0x48) has 3 bits in it called “out DMA state”. Can I look at it alone to determine whether the current chain is complete?
A: No. There is no way to look at the DMA state alone to determine whether the current chain is completed. The state machine returns 0 every time a data block is complete and checks whether the next DT or next Linked List is valid. If invalid, the state machine stops there; else it starts to consume the next DT block. To determine whether the chain is complete you need to make sure of the following: (a) the current Data block is complete by checking status bit or DMA state=0, and (b) next DT is invalid. Unfortunately there is no quicker way. The state machine runs as follows: 0 (idle) -> 1 (descriptor table load) -> 2 (Data request) -> 3 (data load) -> 4 (status bit update) -> 0 (idle).

Q: What’s the relation between Register (0x48) “Output DMA DT Size and State” and register (0x8C) “Output DMA Fifo Info”?
A: Originally there was only reg0x48, no reg0x8C. Later the decision was made to increase the size of FIFO to 90 deep. As a result, reg0x48 became insufficient. Its fields “out FIFO Used” and “out FIFO Free” were not wide enough to express the number 90. So the decision was made to move the two fields to the new register 0x8C.

# Appendix B: Revision History

| Date | Rev. | Comment |
|---|---|---|
| September, 2008 | 3.03 | First release of the public version. |